SÍNTESE DE PERFORMANCES REALÍSTICAS DE DANÇA CONDICIONADA A DADOS MUSICAIS UTILIZANDO REDES CONVOLUCIONAIS EM GRAFOS

JOÃO PEDRO MOREIRA FERREIRA

SÍNTESE DE PERFORMANCES REALÍSTICAS DE DANÇA CONDICIONADA A DADOS MUSICAIS UTILIZANDO REDES CONVOLUCIONAIS EM GRAFOS

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Orientador: Erickson Rangel do Nascimento
Coorientador: Renato José Martins

Belo Horizonte/MG

Outubro de 2020

JOÃO PEDRO MOREIRA FERREIRA

SYNTHESIZING REALISTIC HUMAN DANCE MOTIONS CONDITIONED BY MUSICAL DATA USING GRAPH CONVOLUTIONAL NETWORKS

Thesis presented to the Graduate Program in Computer Science of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Master in Computer Science.

Advisor: Erickson Rangel do Nascimento
Co-Advisor: Renato José Martins

Belo Horizonte/MG
October 2020

Ferreira, João Pedro Moreira.

F383s Synthesizing realistic human dance motions conditioned by musical data using graph convolutional networks [manuscrito] / João Pedro Moreira Ferreira. –
2020.
xv, 65 f. il.

Orientador: Erickson Rangel do Nascimento.
Coorientador: Renato José Martins
Dissertação (mestrado) - Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação.
Referências: f.59-65

1. Computação – Teses. 2. Movimento humano –Teses. 3. Visão por computador – Teses. 4. Redes neurais convolucionais – Teses. I. Nascimento, Erickson Rangel do. II. Martins, Renato José III. Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação. III.Título.

CDU 519.6*82(043)

Ficha catalográfica elaborada pela bibliotecária Belkiz Inez Rezende Costa
CRB 6ª Região nº 1510

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Synthesizing Realistic Human Dance Motions Conditioned by Musical Data using Graph Convolutional Networks

JOÃO PEDRO MOREIRA FERREIRA

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

Prof. Erickson Rangel do Nascimento - Orientador
Departamento de Ciência da Computação - UFMG

Dr. Renato José Martins - Coorientador
Pós-Doutorando - INRIA

Prof. Diego Roberto Colombo Dias
Departamento de Ciência da Computação - UFSJ

Prof. Marcos de Oliveira Lage Ferreira
Instituto de Computação - UFF

Prof. Mario Fernando Montenegro Campos
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 30 de Outubro de 2020.

Este trabalho é dedicado a todas as pessoas que possuem por mim algum afeto.

Acknowledgments

Primeiramente agradeço a Deus, sem ele nada seria possível. Agradeço toda a minha família, e em especial meus pais e minha irmã que sempre dedicaram o melhor de seus esforços pelo meu bem. Agradeço também aos professores que fizeram parte da minha formação acadêmica, sejam eles do primário ou das últimas disciplinas da graduação ou da pós. Nesse contexto gostaria de ressaltar a gratidão ao professores Erickson e Renato que me orientaram neste trabalho, apesar das falhas do percurso, tenho como extremamente positiva essa experiência em minha vida. Aos meus amigos que sempre estiveram comigo tanto nos momentos ruins quanto nos bons. Aos colegas de curso pelo convívio durante esse tempo de pós graduação. Agradeço em especial ao amigo de jornada enquanto no VeRLab, Thiago Luange, que durante os últimos dois anos foi mais que um colega de jornada. Aos servidores da Universidade que sempre que necessário foram extremamente prestativos comigo. Agradeço o fomento para pequisa de todas as agências em especial a FAPEMIG, a CNPq, e a CAPES por viabilizar o ecosistema de pesquisa de excelência. A Universidade Federal de Minas Gerais pelas oportunidades oferecidas, pelos amigos encontrados, pelos momentos vividos, e principalmente pelo amadurecimento como pessoa. Ao VeRLab pelas amizades feitas, e convívio intenso durante essa jornada.

Enfim a todos que contribuíram de alguma forma em minha vida, meu mais sincero.

Obrigado!

“...man, in his quest for knowledge and progress, is determined and cannot be deterred.”

(John F. Kennedy)

Resumo

A síntese de movimento humano utilizando técnicas de aprendizado de máquina tem se tornado cada vez mais promissora para reduzir a necessidade de captura de dados para a produção de animações. Aprender a mover-se de maneira natural a partir de um áudio, e particularmente aprender a dançar, é uma tarefa difícil que humanos frequentemente realizam com pouco esforço. Cada movimento de dança é único, mas ainda assim esses movimentos preservam as principais características do estilo de dança. A maioria das abordagens existentes para o problema de síntese de dança utiliza redes convolucionais clássicas e redes neurais recursivas no processo de aprendizagem. No entanto, elas enfrentam problemas no treinamento e na variabilidade dos resultados devido à geometria não Euclideana da estrutura da variedade do espaco de movimento. Nesta dissertação é proposta uma nova abordagem inspirada em redes convolucionais em grafos para tratar o problema de geração automática de dança a partir de áudio. O método proposto utiliza uma estratégia de treinamento adversário condicionada a uma música para sintetizar movimentos naturais preservando movimentos característicos dos diferentes estilos musicais. O método proposto foi avaliado em um estudo de usuário e com três métricas quantitativas, comumente empregadas para avaliar modelos generativos. Os resultados mostram que a abordagem proposta utilizando redes convolucionais em grafos supera o estado da arte em geração de dança condicionada a música em diferentes experimentos. Além disso, o modelo proposto é mais simples, mais fácil de ser treinado, e capaz de gerar movimentos com estilo mais realista baseado em diferentes métricas qualitativas e quantitativas do que o estado da arte. Vale ressaltar que o método proposto apresentou uma qualidade visual nos movimentos gerados comparável a movimentos reais.

Palavras-chave: Geração de movimento humano, Processamento de áudio e dança, Aprendizado multi-modal, Redes adversárias condicionadas, Redes convolucionais em grafos.

Abstract

Synthesizing human motion through learning techniques is becoming an increasingly popular approach to alleviating the requirement of new data capture to produce animations. Learning to move naturally from music, i.e., to dance, is one of the more complex motions humans often perform effortlessly. Each dance movement is unique, yet such movements maintain the core characteristics of the dance style. Most approaches addressing this problem with classical convolutional and recursive neural models undergo training and variability issues due to the non-Euclidean geometry of the motion manifold structure. In this thesis, we design a novel method based on graph convolutional networks to tackle the problem of automatic dance generation from audio information. Our method uses an adversarial learning scheme conditioned on the input music audios to create natural motions preserving the key movements of different music styles. We evaluate our method with three quantitative metrics of generative methods and a user study. The results suggest that the proposed GCN model outperforms the state-of-the-art dance generation method conditioned on music in different experiments. Moreover, our graph-convolutional approach is simpler, easier to be trained, and capable of generating more realistic motion styles regarding qualitative and different quantitative metrics. It also presented a visual movement perceptual quality comparable to real motion data.

Keywords: Human motion generation, Sound and dance processing, Multi-modal learning, Conditional adversarial nets, Graph convolutional neural networks.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

One of the enduring grand challenges in computer graphics is to provide plausible animations to virtual avatars. Humans have a rich set of different movements when performing activities such as walking, running, jumping, or dancing to music. Over the past several decades, modeling such movements has been relegated to motion capture systems. Despite remarkable results achieved by highly skilled artists using captured data, the human motion has a rich spatiotemporal distribution with an endless variety of different motions. Moreover, human motion is affected by complex situation-aware aspects, including the auditory perception, physical conditions such as the person's age and its gender, and cultural background.

Synthesizing motions through learning techniques is becoming an increasingly popular approach to alleviating the requirement for new data capture to produce animations. The motion synthesis has been applied to a myriad of applications such as graphic animation for entertainment, robotics, and in multimodal graphic rendering engines with human crowds as presented by Ikeuchi et al. [2018], to name a few. Movements of a human being can be considered unique having its particularities, yet such movements preserve the characteristics of the motion style (*e.g.*, walking, jumping, or dancing), and we are often capable of identifying the style effortlessly. When animating a virtual avatar, the ultimate goal is not only retargeting a movement from a real human to a virtual character but embodying motions that resemble the original human motion. In other words, a crucial step to achieve plausible animation is to learn the motion distribution and then draw samples (*i.e.*, motions) from it. For instance, a challenging human movement is dancing, where the animator does not aim to copy and paste poses into the avatar skeleton but to generate a set of poses that match the music's choreography, preserving the quality of being individual.

Moreover, in the last years, we witness the use of multimodal data to improve

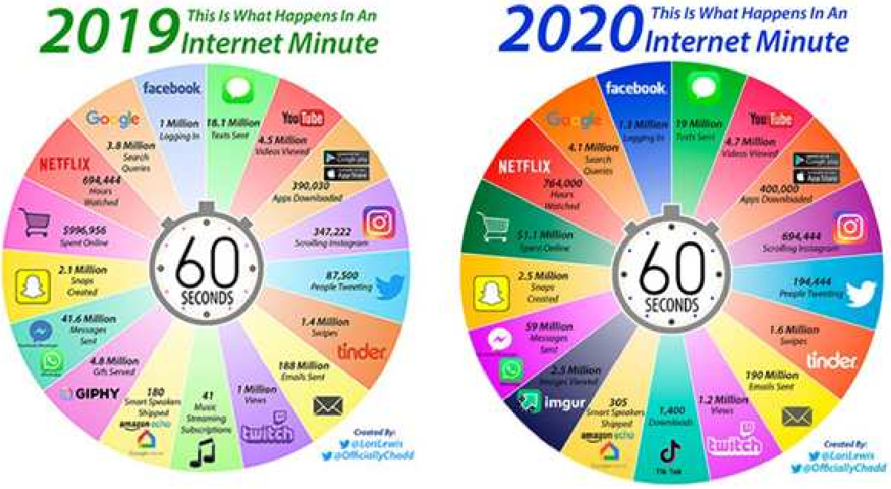


Figure 1.1. The amount of publicly available data on the internet allows the construction of algorithms able to model a distribution of the data that is closer to the real data. Moreover, the growth over the years further improve the results of data-driven approaches. Image courtesy of cloudnine e-discovery daily[2].

the results of learning-based techniques. In a scene there is visual information that often is related to the audio information, the use of the audio data to help learning techniques to improve their results in commonly visual problems has shown impressive achievements, as the work presented by Aytar et al. [2016]. Together with the advances of the techniques, we observe an explosion in the amount of data publicly available on the internet, the increasing number of content creators [1] yielded the growth of publicly available data on every modality (visual, textual, sound, etc.). Figure 1.1 shows both, the amount of data created over a minute on the internet for several social media, and the growth of the data created on the last year. The algorithms capable of dealing with multimodal information, and the increasing amount of publicly available data on the internet allow us to create solutions based on data-driven approaches, which has shown impressive results in several motion, appearance, and related generation tasks.

Problem: In this thesis, we address the problem of synthesizing dance movements from music using adversarial training and a convolutional graph network architecture (GCN). Dancing is a representative and challenging human motion since dancing is more than just performing pre-defined and organized locomotor movements, but it comprises steps and sequences of self expression. In dance moves, both the particular-

[1] https://medium.com/should-you-consider-becoming-a-content-creator-in-2019-statistics-say-you-should

[2] https://cloudnine.com/ediscoverydaily/electronic-discovery/no-fooling-its-time-for-the-2020-internet-minute-infographic-ediscovery-trends/

ities of a person performing the choreography and the characteristics of the movement play an essential role in recognizing the dance style. Thus, a central challenge in our work is to synthesize a set of poses taking into account three main aspects:

1. The motion must be plausible, *i.e.*, a blind evaluation should present similar results when compared to real motions;
2. The synthesized motion must retain all the characteristics present in a typical performance of the music's choreography;
3. Each new set of poses should not be strictly equal to another set, *i.e.*, when generating a movement for a new avatar, we must retain the quality of being individual.

Creating motions from sound relates to the paradigm of embodied music cognition. It couples perception and action, physical environment conditions, and subjective user experiences (cultural heritage) as addressed by Leman [2014]. Moreover, creating realistic motions to be used in animations to virtual avatars become even a harder problem when we aim to create realistic animations according to a sound, as the authors of Shlizerman et al. [2018] explored in their work. Therefore, synthesizing realistic human motions regarding embodying motion aspects remains as a challenging and active research field with recent works as Ginosar et al. [2019] and Yan et al. [2019]. Modeling distributions over movements is a powerful tool that can help us to create a large variety of motions while not removing the individual characteristics of each sample that is drawn. Furthermore, by conditioning these distributions, for instance, using an audio signal like music, we are able to select a sub-population of movements that match with the input signal.

Generative models have demonstrated impressive results in learning data distribution. These models have been improved over the decades through machine learning advances that broadened the understanding of learning models from data. In particular, advances in the deep learning techniques yielded an unprecedented combination of effective and abundant techniques able to predict and generate data. The result was an explosion in highly accurate results in tasks of different fields. The explosion was felt first and foremost in the computer vision community: from high accuracy scores in image classification using convolutional neural networks (CNN) to photo-realistic image generation provided by generative adversarial networks (GAN) proposed by Goodfellow et al. [2014], computer vision field has been benefited with several improvements in the deep learning methods. It is worth noting that the computer vision and computer graphics fields achieved great advances in processing multimodal data present in the scene by using several types of sensors. These advances are assigned to the recent rise of learning approaches to process signal data, especially the convolutional neural networks. Moreover, these approaches have been explored to synthesize data from multimodal sources. And the audio data is one that is achieving the most impressive results, as the work presented by Cudeiro et al. [2019]. However, when we aim to model the distribution of the human motion, the manifold structure poses several issues to traditional CNN models.

Most recently, networks operating on graphs have emerged as promising and effective approaches to deal with the tasks when the structure is known *a priori*. A representative approach is the work of Kipf and Welling [2017], where a convolutional architecture that operates directly on graph-structured data is used in a semi-supervised classification task. Since graphs are natural representations for the human skeleton, several approaches using graph convolutional networks (GCN) have been proposed in the literature to estimate and generate human motion. The work of Yan et al. [2019], for instance, presented a framework based on GCNs. The authors' framework generates a set of skeleton poses by sampling random vectors from a Gaussian process (GP). Despite being able to create sets of poses that mimic a person dancing, the framework does not provide any control over the rhythm or dance style, in other words there is no mechanism to condition the motion generation. Our methodology to synthesize human movements also makes use of GCN, but unlikely the work of Yan et al. [2019], we can control the dance style using audio data while preserving the plausibility of the final motions. We argue that human movements, as having a graph-structured model, follow complex sequences of poses that are temporal related, and the set of defined and organized locomotions can be better modeled using a convolutional graph network trained using adversarial regime.

In this context, we propose in this thesis an architecture that is able to manage multimodal data to synthesize motion, especially an architecture capable of managing audio data to synthesize motion data. Our method starts encoding a sound signal and extracts the style using a CNN architecture. This architecture maps a sound signal to a dense representation of the audio class and generates a spatial-temporal latent vector, conditioned on the dance style. The music style provides control to the motion generation architecture that is based on a graph convolutional neural network. This second architecture is trained in an adversarial regime and predicts the 2D human body joint positions over time. Experiments with a user study and quantitative metrics showed that our method outperforms the state-of-the-art method and provides plausible movements while maintaining the characteristics of different dance styles.

Thesis Statement: We argue that the auditory information (from music) is closely related to dance styles movements of people on videos (motion). Thus, we propose to explore this relationship between auditory and visual data to synthesize realistic human dance performances.

Contributions: The main contributions of this thesis can be summarized as follows:

- A new conditional GCN architecture to synthesize human motion based on auditory data. In our method, we push further the adversarial learning to provide a motion generation algorithm conditioned on the audio information;
- A novel multi-modal dataset. The dataset comprises audio and visual motion data, and the motion data is represented as a set of temporally coherent human poses.

The proposed method outperforms the state-of-the-art in human dance generation from audio, achieving in a blind user study scores similar to real movements. The results achieved by the method in the thesis resulted in a paper accepted for publication in the Computers & Graphics journal, Ferreira et al. [2020].

This thesis is organized as follows: ***i)*** The Chapter 2 introduces some discussions and gives background information to familiarize the reader with the terminology and motion generation techniques are further discussed. It also presents related works in the field and discuss their approaches, results, and contributions. Finally, some previous modelling attempts are briefly described with their results; ***ii)*** In Chapter 3 we present our proposed approach to handle the problem, and explain the details about our new conditional GCN architecture; ***iii)*** In Chapter 4 we introduce our novel multi-modal dataset, we also clarify the pipeline used to collect and prepare the data; ***iv)*** Then we present our experiments protocol together with our results in Chapter 5; ***v)*** We present our conclusions on Chapter 6 with future research directions that can be explored to improve our work.

Chapter 2

Background & Related Work

In this Chapter we introduce some of the techniques we adopted to build our motion generation approach. Also, this Chapter aims to present some of the terms we will use during this thesis, and bring the reader closer to them. In addition to that, we present closely related problems that paved the way to our method of motion generation based on auditory data. Moreover, we present a discussion of our previous attempts to address the problem, with the insights we had to improve the results.

2.1 Theoretical Background

In this section we present some formulations that are adopted to design our approach. Together with these formulations, the literature terminology is presented with some illustrations.

2.1.1 Graph Convolutional Layers

Graph Convolutional Networks (GCN) are becoming in the last years a suitable mechanism to create deep learning models using data structured as graphs, as social networks and protein-interaction networks for example. Since Kipf and Welling [2017] formulated in their work a method to construct graph convolutional networks, several tasks improved results following their formulation, as an example the action recognition task.

To create their formulation Kipf and Welling [2017] assumes an information propagation rule defined as:

$$f(H^{(l)}, A) = \sigma(AH^{(l)}W^{(l)}), \tag{2.1}$$

where σ is a non-linear activation function, A is the adjacency matrix , matrix which defines the conections between the nodes in the graph, $H^{(l)}$ are the features in the l-th layer, and $W^{(l)}$ are the weights of the l-th layer. Thus, with the propagation function defined in Equation 2.1 we can also build classic convolutional neural networks (CNN models), in addition to GCNs.

With the formulation of the propagation function described in Equation 2.1, we can define a propagation rule that can be used in graphs, by changing the adjacency matrix to represent a graph. However, two major issues raise when using the adjacency matrix of a graph. First, real problems modeled with graphs generally do not have self-loops, for example, in a social media network like Twitter, you cannot follow yourself, or on Facebook, you cannot be a friend of yourself, thus creates an issue since features of a node in a graph will normally not be propagated to itself. The second major issue is that the adjacency matrix is typically not normalized, most graph topologies are not as well structured as the image topology (pixels neighborhood). Thus, the usage of the propagation rule defined in Equation 2.1 can create scale problems in the features.

Both major limitations were addressed by Kipf and Welling [2017] in their work. The formulation they proposed follows a two-step process, to address both challenges. First, to deal with the problem of nodes receiving features of them during the propagation process, an addition between the adjacency matrix and an identity matrix is done, creating self-loops or enhancing them if they already exist in the graph. Second, to address the scale problems due to the not normalized adjacency matrix, a symmetric normalization, that averages the adjacency matrix nodes' neighbourhood is applied. With this, the formulation proposed by Kipf and Welling [2017] can be defined as:

$$f(H^{(l)}, A) = \sigma(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}), \tag{2.2}$$

where $\sigma(\cdot)$ is a non-linear activation function, like LeakyRelu, $\hat{A}$ is the result of sum between the adjacency matrix and the identity matrix, the mechanism to address the first issue, $\hat{D}$ is the diagonal node degree matrix used to normalize the adjacency matrix, used in the symmetric normalization of $\hat{A}$, addressing the second issue. $\hat{W}$ is the weight matrix of the layer and H are the values of the features for each layer.

In practice, as addressed by Wu et al. [2020] an image can be considered a special case of graphs, where the adjacency of pixels, *i.e.*, their adjacency matrix, is well defined and known. This relationship between images and graphs can be seen in Figure 2.1. To address our problem and to fulfill the restrictions inherent to the manifold of the human body motion, we construct our model architecture using graph convolutional layers.

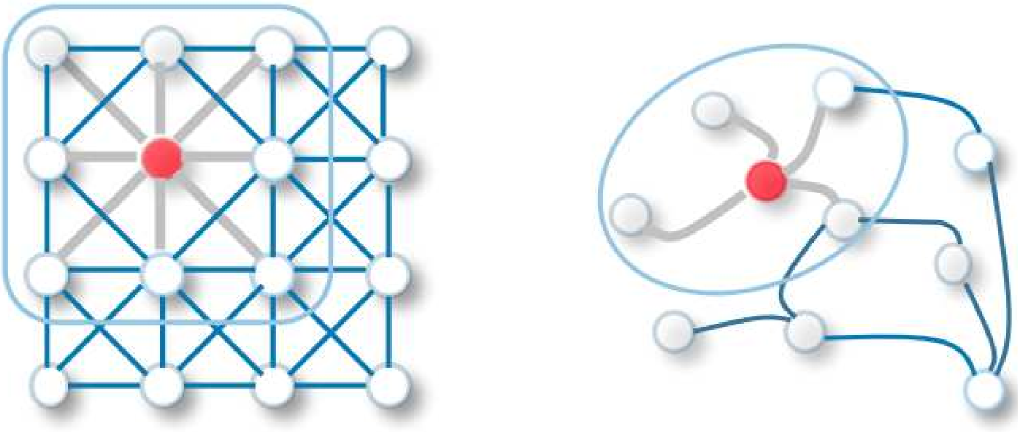


Figure 2.1. Illustration of a classic two-dimensional convolution and a graph convolution. In left an illustration of standard two-dimensional convolution operation, and in the right an illustration of a graph convolution operation. Image courtesy of Wu et al. [2020].

2.1.2 Generative Models

The goal of generative models is, given a set of training data, generate new samples from the same distribution. In other words, generative models address the density estimation problem, which is try to model the probability density function based on observed data. Generative models can be used in several applications, such as super-resolution, colorization, and content generation (as in our case).

Although there is a set of generative models and each one of them with its pros and cons, one of them has been widely adopted by the community. This generative model is GAN's (Generative Adversarial Networks), in which we are particularly interested in one of its variations, cGAN's (Conditional Generative Adversarial Networks). GAN's were first introduced by the breakthrough work of Goodfellow et al. [2014], and cGAN's were introduced latter on by Mirza and Osindero [2014]. GAN's are networks that learn to sample from a distribution of data. To do so, we train two networks to compete against each other, looking towards a Nash equilibrium. The network which learns to sample from the distribution is called generator, and the network competing against the generator is usually called discriminator. The discriminator network aims to distinguish the real samples, taken from the training set, from the fake ones, generated by the generator network. cGAN's in another hand, follow the same principle of GAN's, however they have the ability of condition the generated sample. The approach with GAN's and cGAN's is not to explicit model the probability density function of the data, but instead we are looking for the ability to sample from the data. To be

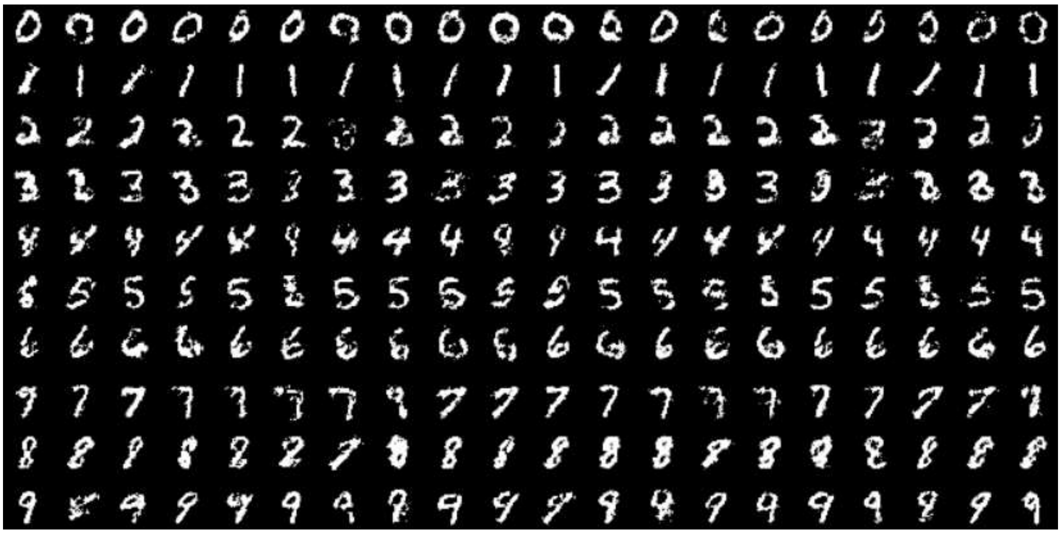


Figure 2.2. Example of cGAN in MNIST dataset, which is a large database of handwritten digits. Each row shows generated samples conditioned by one label (*i.e.* one number between 0 and 9). Image courtesy of Mirza and Osindero [2014].

able of sampling from the data GAN's and cGAN's follow a game-theoretic approach, where two players (generator and discriminator) compete against each other until the generator is able to produce realistic samples. This game-theoretic approach is modeled by Equation 2.3 in GAN's case, and by Equation 3.6 in cGAN's case. Generally GAN's aim to implicit model the probability density function of the dataset distribution. Following Equation 2.3 we have the generator (G), the discriminator (D), a sample of the dataset (x), and finally a random noise (z). The random noise is usually called latent space, the generator uses it to synthesize a fake sample, trying to approximate it to the samples in the real dataset, using a *minmax* loss:

$$\begin{aligned}\min_G \max_D V(D,G) = \mathbb{E}_{x \sim p_{data}}(x)[\log D(x)]+ \\ +\mathbb{E}_{z \sim p_z}(z)[\log(1-D(G(z)))]\end{aligned} \tag{2.3}$$

The results GAN's and cGAN's can achieve jointly made them a suitable approach for our problem. In our problem, we seek to generate realistic samples of human motion conditioned on auditory data, which perfectly fits with cGAN's techniques. The realism of the samples generated using these techniques, and the ability of cGAN's to condition the sampling process are impressive, and can be seen in Figure 2.2. However, as stated in literature using GAN's or cGAN's with others classical loss functions can improve the results, an important work of the area that stated this is the work of Wang et al. [2018].

2.1.3 Human Motion Estimation

Human Motion Estimation is a widely studied problem in the computer vision community. Estimate the motion can be useful in various scenarios, *e.g.* Human Machine Interface, Realistic Animation of Virtual Avatars, and Sports Analytics to name a few. Most of the methods for human motion estimation are based on estimate human pose throughout time. We found in literature methods for human pose estimation in $2D$, $2.5D$, and $3D$, all of them can be adapted to be used in human motion estimation problem. The community also started to step forward in methods considering the temporal constrains of the problem of human motion estimation. An notorious example of methods specifically designed for human motion estimation is the work of Kocabas et al. [2020]. In this thesis, as we are interested in generating human motion, these methods for human pose estimation suit as automatic data annotators.

Despite the fact that, the human motion is for definition in a three-dimensional space, the data (*i.e.* images) we use to infer it are essentially in a two-dimensional space. It is possible to infer the human pose and shape, and go beyond to infer the human motion in a three-dimensional space. Several works address the problem of human pose and shape estimation in a three-dimensional space, *e.g.* Kanazawa et al. [2019], and Kolotouros et al. [2019]. Moreover, adding temporal constraints to the initial problem to model the human motion in a three-dimensional space we can highlight the work of Kocabas et al. [2020]. An illustration of these methods are shown in Figure 2.3. Nevertheless, these methods have some drawbacks to fit in the context of an automatic data annotator for our task of human motion generation. The two major drawbacks are: *i)* The three-dimensional techniques are more sensible to noise than the two-dimensional ones; *ii)* The $3D$ representation for human body and shape are way more complex than $2D$ representations. Thus, both these drawbacks increase our problem complexity, on one hand adding more noise to our dataset, and on the other hand learning a data structure more complex than we need to tackle the problem.

Also, the data structure used to represent the human pose, and over time the human motion in a $2D$ space is remarkably simpler. There was a notably effort of the community in the last few years, looking towards the improvement of the results for the human pose estimation problem in the wild Wei et al. [2016]; Cao et al. [2017]; Simon et al. [2017]; Fang et al. [2017]; Li et al. [2019]; Xiu et al. [2018]; Cao et al. [2019]. There are even methods to estimate a dense representation of the human body in the image space, thus resulting in a $2.5D$ representation of the human body pose, and human motion Güler et al. [2018]. Most of these methods were implemented in challenges motivated by the industry Lin et al. [2014]. Moreover, most of these methods has

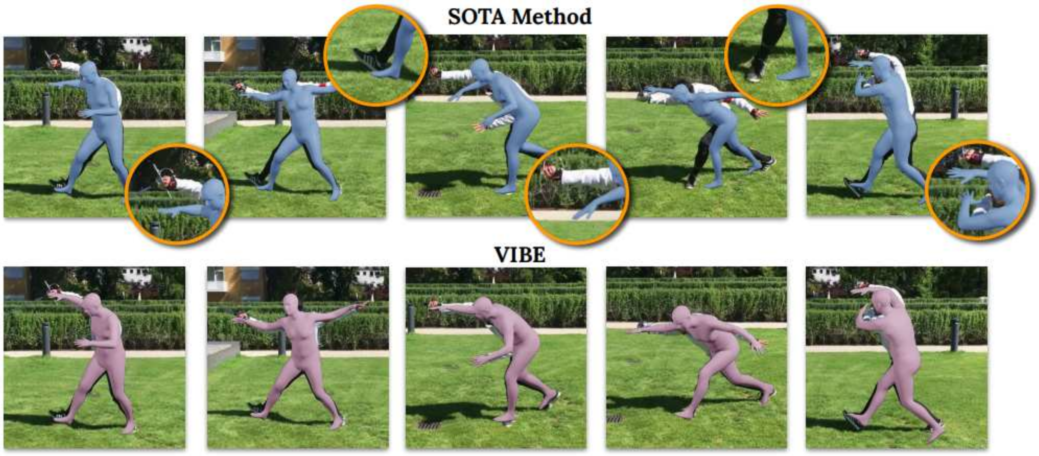


Figure 2.3. Comparison between a temporal aware method for $3D$ human motion estimation and a state-of-the-art method for $3D$ human pose and shape estimation. Image courtesy of Kocabas et al. [2020]

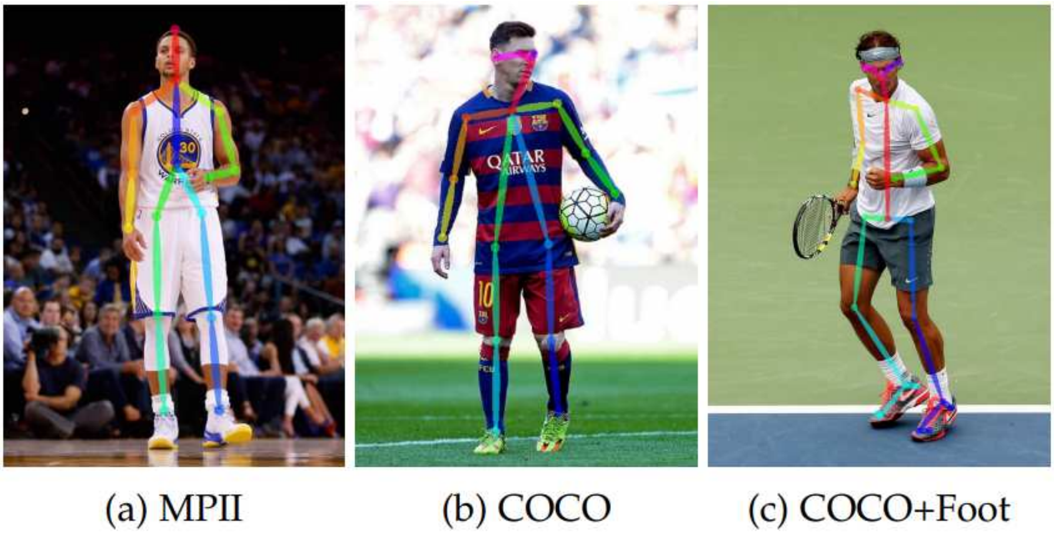


Figure 2.4. OpenPose framework results on three different commonly used datasets standards. Image courtesy of Cao et al. [2019]

been continuously improved over time, with new features, or new datasets standards, , as shown in Figure 2.4. In this thesis, we choose the state-of-the-art work of Cao et al. [2019], entitled OpenPose framework, because of its accurate results, maximal compatibility with existing datasets, and for being a two-dimensional technique.

2.2 Related Work

Techniques based on data-driven methods using multimodal data are becoming a clever way to explore the increasing amount of publicly available data on the internet, to improve the results of non-multimodal approaches. In the following sections we present some remarkable works in the related literature of motion generation based on auditory data.

2.2.1 Sound and Motion

Recently, we have witnessed an overwhelming growth of new approaches to deal with the tasks of transferring motion style and building animations of people from sound. Early methods were based on editing existing videos by selecting the visual segments that match a particular audio. For example, Bregler et al. [1997] create videos of a subject saying a phrase they did not say originally, by reordering the mouth images in the training input video to match the phoneme sequence of the new audio track. In the same direction, Weiss [2005] applied a data-driven audio-visual approach to produce a 2D video-realistic audio-visual "Talking Head", using F0 and Mel-Cepstrum coefficients as acoustical features to model the speech. Aiming to synthesize human motion according to music characteristics such as rhythm, speed, and intensity, Shiratori and Ikeuchi [2008] established keyposes according to changes in the rhythm and body movement (performer's hands, feet, and center of mass), and then used music and motion feature vectors to select candidate motion segments that match music and motion intensity. Despite their impressive results, its method fails when these keyposes are present in fast passages of a music.

Cudeiro et al. [2019] presented an encoder-decoder network that utilizes audio features extracted from DeepSpeech presented by Hannun et al. [2014]. The network generates realistic 3D facial animations conditioned on subject labels to learn different individual speaking styles. To deform the human face mesh, Cudeiro et al., similarly to our work, encode the audio features in a low-dimensional embedding space. Although their model is capable of generalizing facial mesh results for unseen subjects, the generated animations are reported to be distant from the natural captured sequences. Also, introducing a new style is cumbersome, since it requires a collection of 4D scans paired with audios. Similarly, Ginosar et al. [2019] enable translation from speech to gesture, generating arms and hands movements by mapping audio to pose. This is accomplished via adversarial training, where a U-Net architecture transforms the encoded audio input into a temporal sequence of 2D poses. In order to produce more realistic results, the discriminator is conditioned on the differences between each pair of subsequent generated poses, but the method is subject specific and does not generalize to other speakers.

Recently, the problem of motion generation, more specifically, the problem of dance generation has been addressed by several works Ren et al. [2019]; Huang et al. [2020]; Li et al. [2020]; Zhuang et al. [2020]; Ye et al. [2020]. The work of Ren et al. [2019] present a method based on RNNs, more precisely, gated recurrent unit (GRU), to deal with temporal constraints of the problem, the authors feed their RNN with an encoded space of the audio data, they also train their method using an adversarial regime, however, their loss function takes into account the activation layers of a GCN, in their case, the ST-GCN proposed by Yan et al. [2018]. Another work addressing the problem of dance generation is the one proposed by Huang et al. [2020]. In their work a music encoder and a dance decoder is used to generate motion, the main difference of their method from the previous methods in the literature for dance generation, is the learning approach. They use a learning strategy called Curriculum Learning proposed by Bengio et al. [2009], which consists of providing the network with easier samples in the beginning of the training stage and then increasing the difficulty of the samples throughout the training stage. Finally, the work of Li et al. [2020] adopted transformer networks, which is a suitable way to treat temporal constraints using self-attention techniques. Differently from most existing approaches, Li et al.'s work can generate 3D dance motions.

Another related work to ours is the approach proposed by Lee et al. [2019]. The authors use a complex architecture to synthesize dance movements (expressed as a sequence of 2D poses) given an input music. Their architecture is elaborated based on a decomposition-to-composition framework trained with an adversarial learning scheme. Our graph-convolutional based approach, on its turn, is simpler, easier to be trained, and generates more realistic motion styles regarding qualitative and different quantitative metrics.

2.2.2 Generative Graph Convolutional Networks

Since the seminal work of Goodfellow et al. [2014], generative adversarial networks (GAN) propose a new technique to generative models where two networks compete against each other in a *minimax* game. In the game, one network aims to generate realistic samples to fool the second network. On the other hand the second network aims to be able of distinguish the real samples from the fake ones, generated by the first network. GAN's have been successfully applied to a myriad of hard problems, notably for the synthesis of new information, such as of images Karras et al. [2018], motion Chan et al. [2019], and pose estimation Chen et al. [2017], to name a few.

Mirza and Osindero [2014] take a step further from the traditional GANs presented by Goodfellow et al., Mirza and Osindero proposed Conditional GANs (cGAN), which provides some guidance into the data generation. Differently from Goodfellow et al.'s work, they were able to control each class of images they aimed to synthesize. Then, improving the technique proposed by Mirza and Osindero [2014] of cGANs, the work proposed by Reed et al. [2016] demonstrated the possibility to use multi-modal data to conditional generative adversarial networks, specifically textual information. In their work, they synthesize realistic images, in other words, given a sentence describing a scene, their method can produce a realistic image that fits the description of the input sentence. Their result demonstrates that cGANs can also be used to tackle multi-modal problems.

Graph Convolutional Networks (GCN) recently emerged as a powerful tool for learning from data by leveraging geometric properties that are embedded beyond n-dimensional Euclidean vector spaces, such as graphs. This was addressed by Kipf and Welling [2017], where a semi-supervised classification on a complex network is performed, achieving impressive results. The work of Jain et al. [2016] uses a recurrent neural network (RNN) as a mechanism to solve temporal issues inherent to the problem, however, the method proposed by Jain et al. [2016] uses spatial-temporal graphs, showing that introducing the data structure of the problem to the learning pipeline can improve the results. In our context, conversely to classical convolutional neural networks (CNNs), GCNs are capable of modeling the motion manifold space structure as shown in the works of Yan et al. [2018]; Yan et al. [2019].

Yan et al. [2018] applied GCNs to model the movement of the human body and classify actions. After extracting 2D human body poses for each frame from the input video, the skeletons are processed by a Spatial-Temporal Graph Convolutional Network (ST-GCN). Their architecture combines the graphs (2D human body poses) information in both, the temporal and spatial dimensions, making their architecture suitable to deal with human motion, which has temporal constraints. Therefore, we also adopt their ST-GCN layer in our GCN architecture. Yan et al. go forward in the representation power of GCNs in Yan et al. [2019] the Convolutional Sequence Generation Network (CSGN). By sampling correlated latent vectors from a Gaussian process and using temporal convolutions, the CSGN architecture was capable of generating temporal coherent long human body action sequences as skeleton graphs. They showed the effectiveness of their method in both 2D and 3D human motion. Our method goes one step further than Yan et al. [2018]; Yan et al. [2019]. It generates human skeletal-based graph motion sequences conditioned on acoustic data, *i.e.*, music. By conditioning the movement distributions, our method learns not only creating plausible human motion but also it learns the music style signature movements from different domains.

2.2.3 Estimating and Forecasting Human Pose

Motion synthesis and motion analysis have been benefited from the impressive improvements in the accuracy of the human pose estimation methods. Human pose estimation from images, for its turn, greatly benefited from the recent emergence of large datasets Lin et al. [2014]; Andriluka et al. [2014]; Güler et al. [2018] with annotated joints positions, and dense correspondences from a 2D image to a 3D human shape. This increase in the amount of available data, and the investment of players of the industry[1] led to a variety of methods, each one with your advantages and disadvantages, some of these works are Cao et al. [2019]; Li et al. [2019]; Xiu et al. [2018]; Güler et al. [2018]; Kolotouros et al. [2019]; Kanazawa et al. [2019]; Kocabas et al. [2020]. The work of Cao et al. [2019] is a CNN that uses the human body structure to process an image and find individually parts of the human body, then combines it to predict the human body pose in the input image. Similar to the work of Cao et al. [2019], the work proposed by Xiu et al. [2018] predicts 2D human body pose, however they go forward on tracking the human movements throughout a video sequence. The work of Li et al. [2019] improves the results of the human pose estimators in a scene with multiple persons, where miss detections are common, they solve the problem by modeling the person nodes, and solving an optimization problem. Güler et al. [2018] add more information to the human pose estimators, by estimating a 2.5D representation of the human pose, where a dense correspondence for a person is given. The works of Kolotouros et al. [2019]; Kanazawa et al. [2019]; Kocabas et al. [2020] estimates the 3D pose and shape of each person. The main difference between these methods is that the method proposed by Kocabas et al. [2020] works on video sequences, while Kolotouros et al. [2019] and Kanazawa et al. [2019] work on each image individually.

This large amount of annotated data made possible important milestones towards predicting and modeling human motions as the works of Wang et al. [2014]; Fragkiadaki et al. [2015]; Ghosh et al. [2017]; Gui et al. [2018] and Wang et al. [2019]. Fragkiadaki et al. [2015], for instance, proposed a recurrent autoencoder model for recognition and prediction of human body pose from videos and motion capture systems. The recent trend in time-series prediction with recurrent neural networks (RNN) has been applied in several frameworks for human motion prediction besides the work

[1] https://cocodataset.org/#keypoints-2016

of Fragkiadaki et al. [2015], the works of Martinez et al. [2017] and Ghosh et al. [2017] also address the same problem. Nevertheless, the pose error accumulation in the predictions allows mostly predicting over a limited range of future frames as addresed by Gui et al. [2018]. Gui et al. [2018] proposed to overcome this issue by applying adversarial training using two global recurrent discriminators that simultaneously validate the sequence-level plausibility of the prediction and its coherence with the input sequence. Wang et al. [2019] proposed a network architecture to model the spatial and temporal variability of motions through a spatial component for feature extraction. Yet, these RNN models are known to be difficult to train and computationally cumbersome as addressed by Pascanu et al. [2013]. Additionally, as also noted by Lee et al. [2019], motions generated by RNNs tend to collapse to certain poses regardless of the inputs.

2.2.4 Transferring Style and Human Motion

Synthesizing motion with specific movement style has been studied in a large body of prior works, as for example, the works of Peng et al. [2018]; Wang et al. [2018]; Kim and Lee [2019]; Smith et al. [2019]; Chan et al. [2019] and Gomes et al. [2020], to name a few. Most methods formulate the problem as transferring a specific motion style to an input motion as done by Xia et al. [2015]; Kim and Lee [2019] and Smith et al. [2019], or transferring the motion from one character to another, commonly referred as motion retargeting as defined by Gleicher [1998] in his seminal work. Other works, following the definition proposed of motion retargeting are the works of Choi and Ko [2000] and Villegas et al. [2018]. Recent approaches explored deep reinforcement learning to model physics-based locomotion with a specific style as done by Peng et al. [2017]; Liu and Hodgins [2018] and Peng et al. [2018].

Another active research direction is transferring motion to video sequences as the work of Wang et al. [2018]; Chan et al. [2019] and Gomes et al. [2020] address in their respective works. However, the generation of stylistic motion from audio is much less explored and it is still a challenging research field. On one hand, the work presented by Wang et al. [2018] is a GAN with a multi-resolution trainning scheme, their framework is generic and the author present results in several contexts, including the rendering of a person that moves accordingly to a motion made by another person. On the other hand, the work proposed by Chan et al. [2019] is a specific framework for video transfer of a person performing a motion, the authors address some issues that the work of Wang et al. [2018] does not, such as improving face details. Finally, the work proposed by Gomes et al. [2020] address the problem of appearance and motion retargeting between videos of different characters with a model-based transferring approach, which provides two advantages: *i)* Their method does not need a large amount of data to deliver satisfactory results; *ii)* Because the nature of their method, they can treat issues related to motion restrictions (interactions between the actor and the environment) this approach is also shape-aware, which allows handling differences between the actors' shapes can produce unfeasible human-to-object interactions in the resulting video.

Villegas et al. [2017] presented a video generation method based on high-level structure extraction, conditioning new frames creation on how this structure evolves in time, therefore preventing pixel-wise error prediction accumulation. This approach was employed on long-term video prediction of humans performing actions by using 2D human poses as high-level structures. Wang et al. [2020] discussed how adversarial learning could be used to generate human motion from sequence autoencoders, focusing on three tasks: motion synthesis, conditional motion synthesis, and motion style transfer. As our work, their framework enables conditional movement generation according to a style, but there is not multimodality associated with it. Jang and Lee [2020] presented a method inspired by sequence-to-sequence models to generate a motion manifold. As a major drawback, the performance of their method decreases when creating movements longer than 10s, which makes the method inappropriate to generate long sequences. Our approach, on the other hand, is capable of generating long movement sequences conditioned on different music styles, by taking advantage of the adversarial GCN's power to generate new long, yet recognizable, motion sequences.

We outline that from the works dealing with the problem of dance motion generation, as the works presented by Lee et al. [2019]; Ren et al. [2019]; Huang et al. [2020]; Li et al. [2020]; Zhuang et al. [2020] and Ye et al. [2020], we are the only method tackling the problem with graph convolutional networks, an architecture aware of the problem manifold structure. Moreover, our approach as the approaches of Lee et al. [2019] and Ren et al. [2019] can produce video sequences with virtual avatars of an actor performing the synthetic motion, while guiding the motion style. Differently from the works of Huang et al. [2020]; Ye et al. [2020] and Zhuang et al. [2020] who treat the temporal issues of the problem using recurrent models that can be difficult to train, as addressed by Pascanu et al. [2013], we solve temporal issues directly in the latent space sampling.

2.3 Our First Modelling Attempts

In this section we present some our previous modelling attempts that did not present satisfactory results. We describe the issues with each approach, and how these issues drive us to the final method presented in the Chapter 3 of this thesis. It is noteworthy that the attempts here presented could produce the desired results. However, during the research we decide to change the approach, following what we observe from the initial experiments and results, and what the literature present to us that could be used in our problem to improve the results.

2.3.1 Convolutional Neural Networks

Our initial approach was inspired in the seminal work of Aytar et al. [2016], hereinafter entitled SoundNet. The SoundNet work shown to the computer vision community that was possible to use deep convolutional networks to address classical visual tasks using multimodal data, in SoundNet case, auditory data.

The SoundNet model propose to address Object and Scene Recognition problems using the auditory data for each one of them. For the Object Recognition problem, it aims to learn the sound that each object generally make (*e.g.* the sound of a musical instrument). To learn this, the authors use the Kullback-Leibler divergence, presented by Kullback and Leibler [1951] with one deep convolutional network trained in the same task. In that way, the approach learns the distribution of a already trained network in visual data, and learn the relationship between the auditory and visual data in the scene. With this protocol and the paired data (*i.e.* paired visual and auditory data), the SoundNet was able to learn the distribution of the "teacher" network and reproduce the results using only auditory data. In the same way SoundNet was also able to present results for Scene Recognition, another classical visual task. An illustration of the SoundNet architecture is shown in Figure 2.5

In one hand, both tasks SoundNet aims to tackle are essentially classification tasks, our problem in other hand, is essentially a regression task (*i.e.* generate motion in a continuous space). The first attempt during the work of this thesis was to adapt the SoundNet architecture to work as a regression model. We choose this, because the paired data is intrinsic to our problem, we need both, the audio to synthesize the motion and the motion to use as ground truth.

Our modifications consisted of changing the final layer of classification to a regression, where we want the layer to predict the 25 joints of the human body following the standard adopted by the OpenPose framework Cao et al. [2019]. Moreover, we

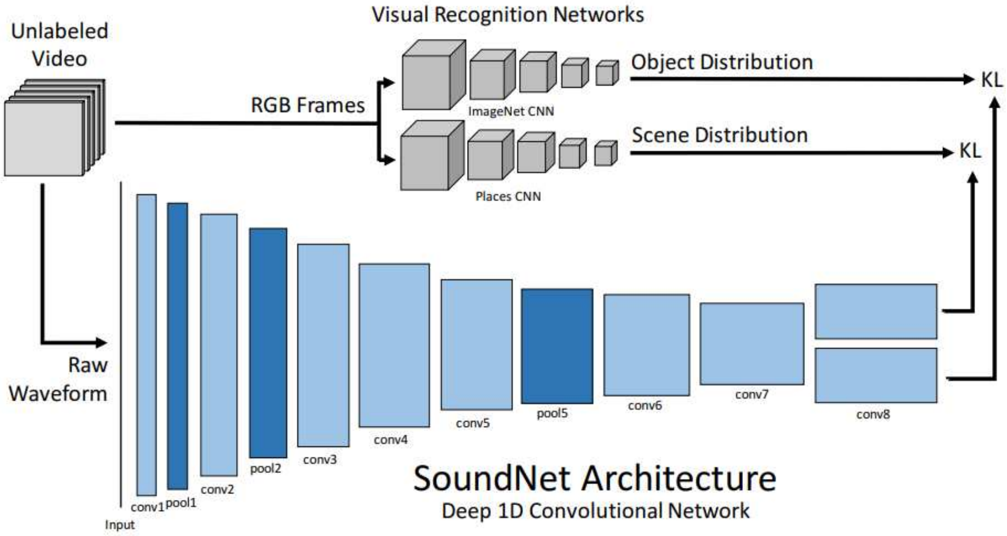


Figure 2.5. SoundNet architecture, a one dimensional CNN model for multi-modal data. SoundNet is able of learn to tackle classical visual tasks (*e.g.* Object and Scene Recognition) using auditory data. Image courtesy of Aytar et al. [2016]

changed the loss, since we do not have a "teacher" network, so we use the $L1$ norm in our experiments. With these modifications, we expected that for each time interval in the auditory data we could predict a human pose using the SoundNet architecture. However, we observed that we need to develop our loss function, and use an adversarial training strategy, because the initial results converged to a mean human pose, where the $L1$ norm assumes the minimum value for the entire dataset. Our discriminator was an vanilla two dimensional convolutional network. Even though we were able to bypass the mean pose issue with the adversarial strategy, this approach using the SoundNet architecture as backbone, presented several issues related to the temporal stability for the human poses. In others words, the temporal constraints of the problem were not respected, since it generated samples with strong variations between the poses in a small time interval.

With the temporal stability of the generated human poses in mind, we decided to tackle the problem using RNN's (Recurrent Neural Networks), since the main goal of these networks is to have a temporal behavior in their outputs. The steps we take in this direction will be presented in the following Section 2.3.2

2.3.2 Recurrent Neural Networks

After the first attempts using convolutional networks failed to address the temporal issues of the problem of human motion generation, we started to explore RNN's

(Recurrent Neural Networks) aiming to circumvent the temporal issues in the initial approach. In our experiments we use LSTM (Long Short Term Memory) and GRU (Gated Recurrent Unit) to construct our model. In our case, the GRU units presented better results, probably because the lower number of parameters.

We start our model based on RNN's following the same principles we use in the first attempt using the SoundNet. We create a RNN using GRU's that receive auditory data as input and predict a human pose following the OpenPose framework convention. We also started our experiments with only the $L1$ norm as loss function. In the initial experiments the model preserved the mean pose prediction, this can be seen in Figure 2.6. However, we observed a second issue, some predictions, especially the first ones did not respect the human body constraints. In other words, our model was predicting human poses that cannot be performed by a human, this can also be seen in Figure 2.6, and in Figure 2.7. In Figure 2.6 we highlight both issues: **a)** a synthetic pose violating the human body constraints; **b)** the mean pose overtime.

Again to tackle the mean pose issue we improved our loss function to consider a adversarial training strategy. We introduced a discriminator network, following the same strategy we adopted in the CNN attempt. Our discriminator was a vanilla two dimensional convolutional network, similar to the one we use in the first attempt. This modification significantly improve our results, some of them can be seen in Figure 2.7. Nonetheless the improvement in the results only regards the mean pose issue, the issue regarding the human body constraints is still present. However, the issue now affects not only the initial poses, but the entire generated movement in some cases. Some of the synthetic poses generated by our RNN model that did not respect the human body constraints are highlighted in Figure 2.7.

Disregarding the issue with the human body constraints, this approach with RNN's begins to deliver promising results. The movements generated by the RNN model trained with an adversarial strategy showed characteristic movements for each dance style, some of them can be seen in Figure 2.7, mainly in the Michael Jackson dance style. These results raised an issue with our dataset. In the first version of our dataset we use the Country dance style. However, as can be seen in the preliminary results shown in Figure 2.7, the Country movements have not enough visual features to be easy to distinguish the dance style in a blind study. This issue with the Country dance style is related with a space normalization we perform in our data. Since the Country initial results were not so promising as the initial results of the other dance style classes, we decided to replace the Country dance style of our dataset by Salsa.

The issue of the predicted poses not respecting the human body constraints shows us that our model architecture, in this case RNN's, were not the most suitable approach

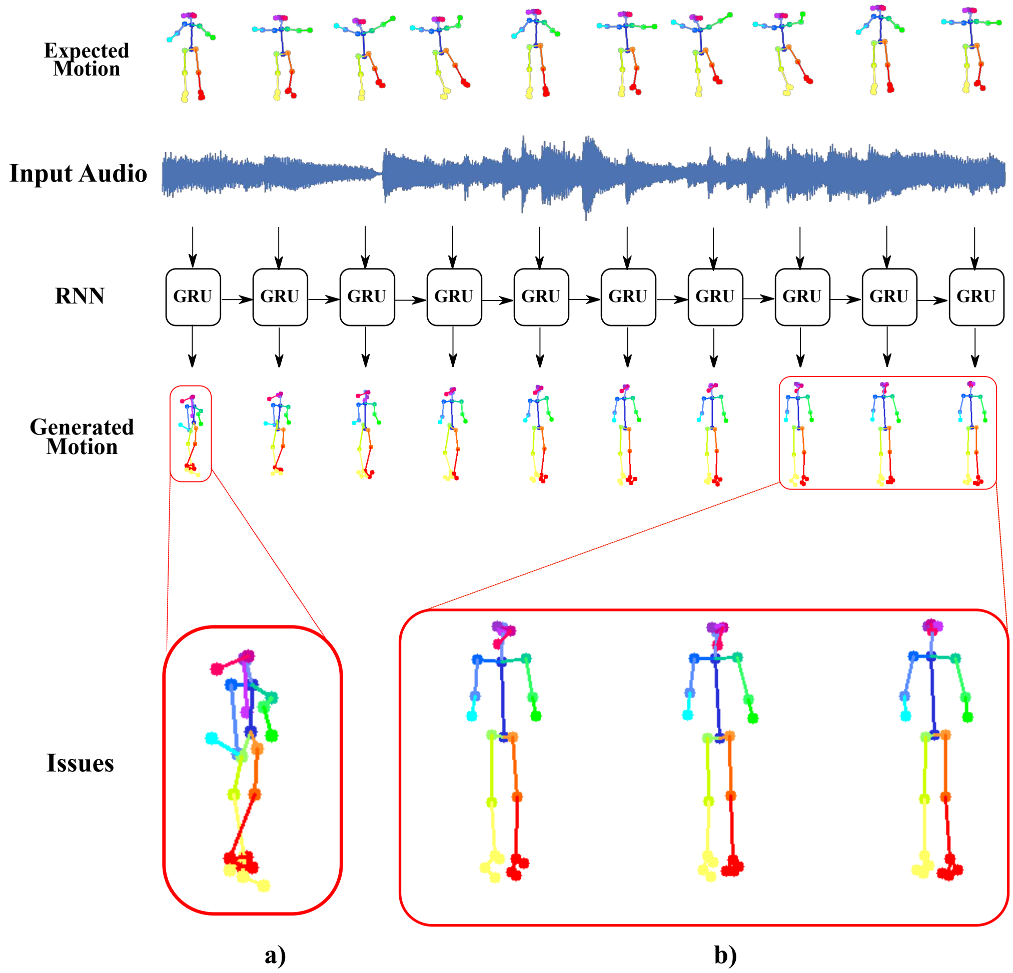


Figure 2.6. One of our first attempts was to use Recurrent Neural Networks. Here are some examples for our model without an adversarial strategy. Nevertheless, two issues arise from this approach: **a)** Some human poses do not respect the human body constraints; **b)** The model converges for a mean human pose, in other words, the motion sequence is static.

for human pose data. The human pose using the OpenPose framework pose convention is essentially a tree (*i.e.* a graph). The intrinsic representation of human pose as a graph, lead us to use GCN's in our approach. GCN's are a recently new class of networks that are presented impressive results in several tasks, because of their ability of modelling the manifold of problems with geometric data structures. We introduce some GCN's and its usage in the previous Section 2.1.1, and we discuss and give the details of our final model using GCN's in the Chapter 3, in which our method is described.

Finally, we would like to emphasizethat although we did not achieve our desired

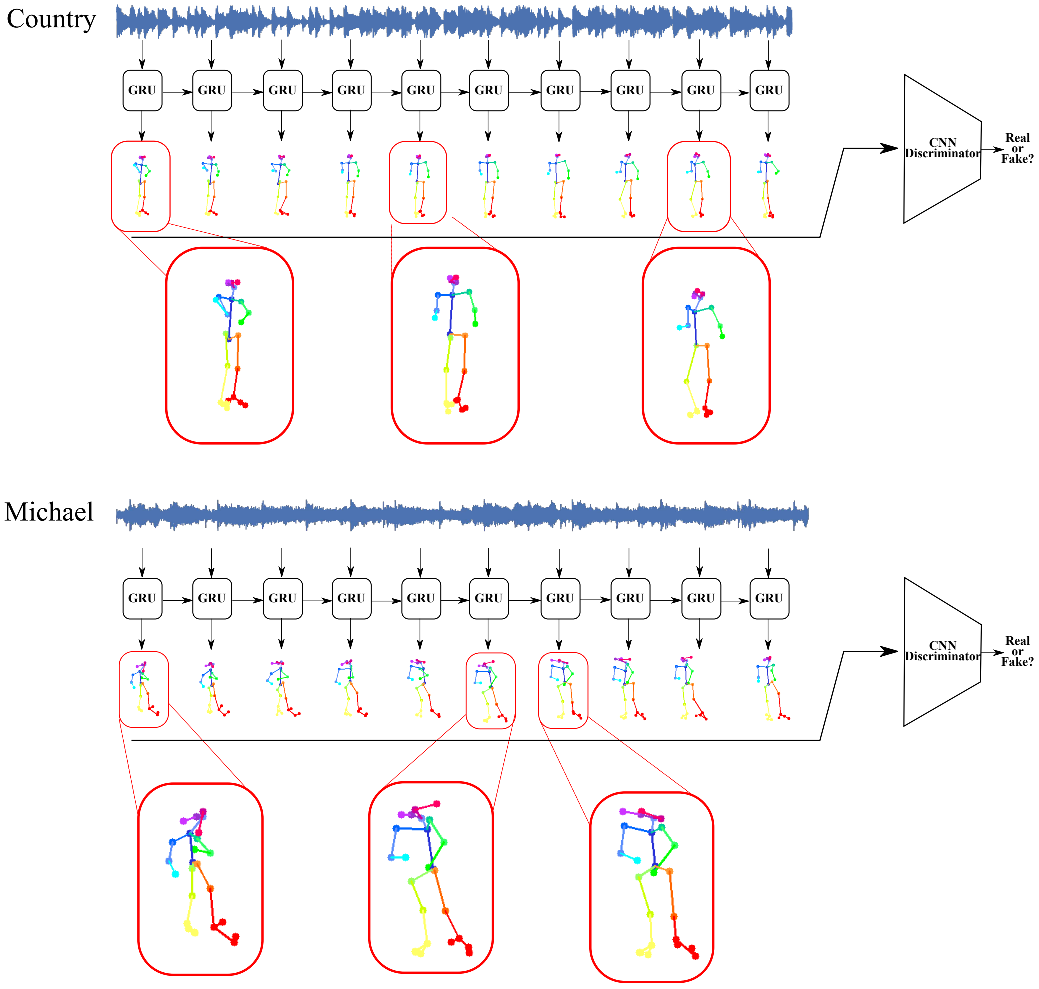


Figure 2.7. Examples for our model with an adversarial strategy. The model was able to learn some movements of the dance style. However, most of the human poses in the sequence remain not respecting the human body constraints.

results using CNN's or RNN's, it can be possible to address the problem of human motion generation using these techniques such as in Wang et al. [2020]; Ren et al. [2019]; Huang et al. [2020]

Chapter 3

Methodology

The problem of automatic dance generation has two major challenges: *i)* The temporal constraint of the human motion; *ii)* The motion manifold, which adds physical constriant's to the joints' spatial configuration of the human body. To address both challenges we develop an architecture based on graph convolutional networks (GCN) since they can be used to both, dealing with the motion manifold and the temporal constraints of the problem.

Moreover, to explore the relationship between auditory and visual data, and to address the lack of visual data in human motion animation, we propose a generative architecture to produce new dance movements. In our architecture, we explore the relationship between auditory and visual data by using the music to condition the generation of our synthetic motion. Thus, our approach aims to address the problem of automatic dance generation, using as input a music song by creating a sequence of human poses and finally creating a video sequence of an actor performing the synthetic motion. An illustration of our proposed approach can be seen in Figure 3.1.

Our method has been designed to synthesize a sequence of 2D human poses resembling a human dancing according to a sound style that is provided by a music as input. Specifically, we aim to estimate a motion $\mathcal{M}$ that provides the best fit for a given input music style. $\mathcal{M}$ is a sequence of N human body poses defined as:

$$\mathcal{M} = [\mathbf{P}_0, \mathbf{P}_1, \cdots, \mathbf{P}_N] \in \mathbb{R}^{N\times 25\times 2}, \tag{3.1}$$

where $\mathbf{P}_t = [\mathbf{J}_0, \mathbf{J}_1, \cdots, \mathbf{J}_{24}]$ is a graph representing the body pose in the frame t and $\mathbf{J}_i \in \mathbb{R}^2$ is the 2D image coordinates of i-th node of this graph (see Figure 3.2).

The advantages of adopting this motion representation are twofold. First, it allows adopting state-of-the-art approaches of human pose estimation to annotate our

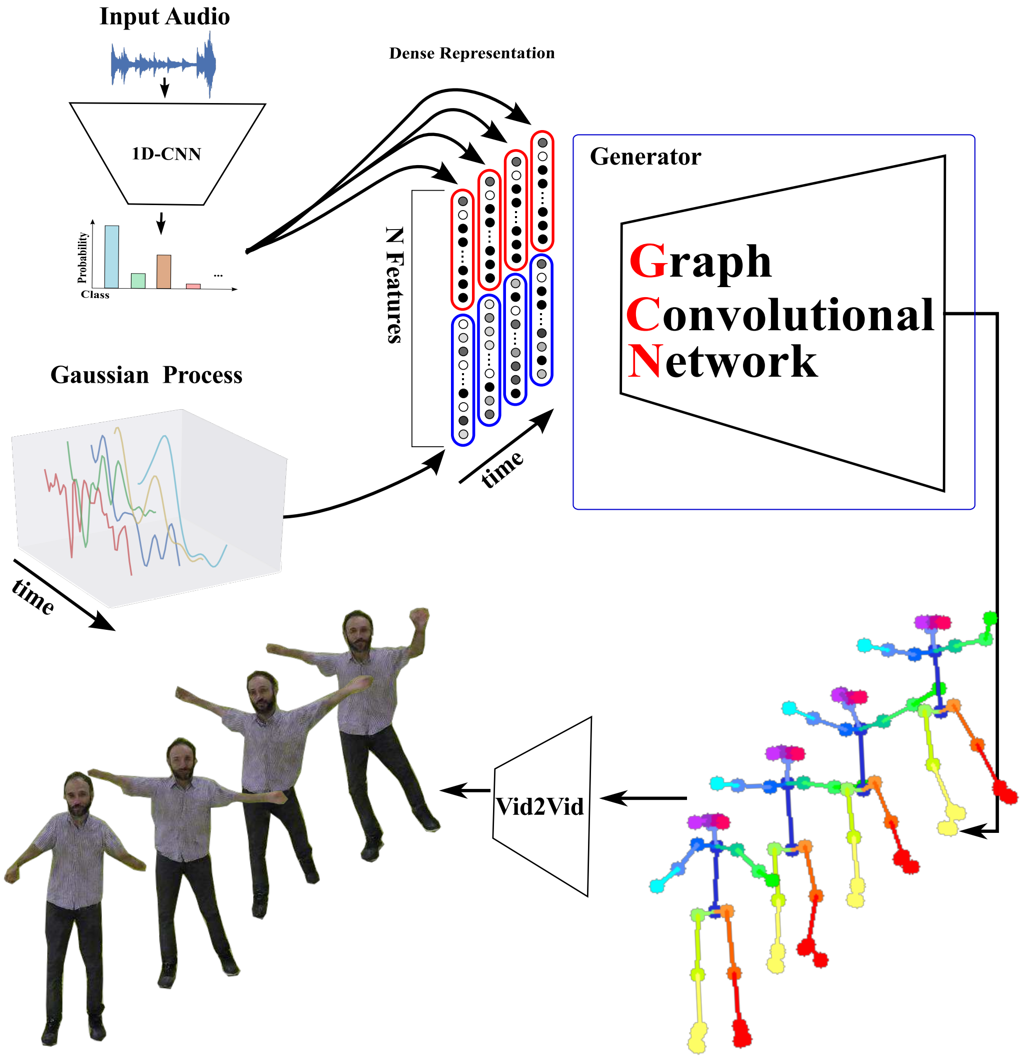


Figure 3.1. Our approach is composed of three main steps: *i)* First, given a music sound as input, we classify the sound to its closest dance style; *ii)* Second, we generate a temporal coherent latent vector to condition the motion generation, *i.e.*, the spatial and temporal position of joints that define the motion. *iii)* Finally, a generative model based on a graph convolutional neural network is trained in an adversarial manner to generate the sequences of human poses. To exemplify an application scenario (and also with visualisation purposes), we also render automatic animations of virtual characters performing the motion generated by our method.

dataset as the work of Cao et al. [2019]. Second, it is also a commonly used representation in motion generation state-of-the-art methods (*e.g.*, Vid2Vid from Wang et al. [2018] or Everybody Dance Now presented by Chan et al. [2019]), which allows evaluations and comparisons in similar conditions.

Our approach consists of three main components, outlined in Figure 3.8. We start

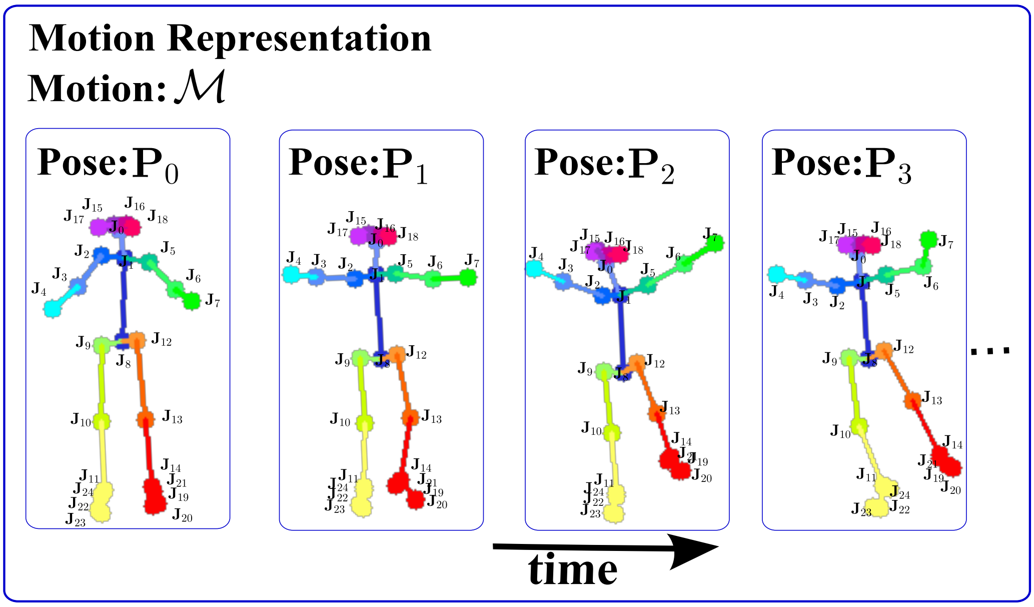


Figure 3.2. Motion and skeleton notations. In our method, we used a skeleton with 25 2D joints.

training a 1D-CNN classifier to define the input music style. Then, the result of the classification is combined with a spatial-temporal correlated latent vector generated by a Gaussian process (GP). The GP allows us to sample points of Gaussian noise from a distribution over functions, where for each function exits a correlation between the sampled points of that function. This correlation can be different for each function sampled by the process. Our goal is to sample temporally correlated points from functions with different frequencies. This variation in the signal frequency enables our model to infer which skeleton joint is responsible for more prolonged movements and explore a large variety of poses. The latent vector aims at maintaining spatial coherence of the motion for each joint over time. At last, we perform the human motion generation from the latent vector. In the training phase of the generator, we use the latent vector to feed a graph convolutional network that is trained in an adversarial regime on the dance style defined by an oracle algorithm. In the test phase, we replace the oracle by the 1D-CNN music classifier. Thus, our approach has two training stages: *i)* The training of the audio classifier to be used in the test phase and *ii)* the GCN training with an adversarial regime that uses the music style to condition the motion generation.

3.1 Sound Processing and Style Feature Extraction

To generate realistic human motion coherent with the input music, we must be able to classify the auditory data. Thus, we address the problem of music classification by training a 1D-CNN classifier. With the music style class, we can condition our motion generation to synthesize a motion sequence coherent with the music style.

Our motion generation is conditioned by a dense representation of the music style in the sound provided by a 1D-CNN classifier presented in Figure 3.3. In this context, we used the architecture of SoundNet proposed by Aytar et al. [2016] as backbone to a one-dimensional CNN. The model receives sound in waveform and outputs the most likely music style. The 1D-CNN classifier is trained in a dataset composed of 107 musics, divided into three music-dance styles: *Ballet*, *Salsa*, and *Michael Jackson (MJ)*.

Our audio classifier architecture is better described in Table 3.1. In general, we use classic one-dimensional convolutions to extract features from the audio data and a fully connected layer in the end to work as a classifier. We standardize the output, to be a vector of probabilities using the SoftMax function. The dimensions of the tensors at each layer are shown in Table 3.1, where the first one is the channels dimension and second one is the length of the sinal.

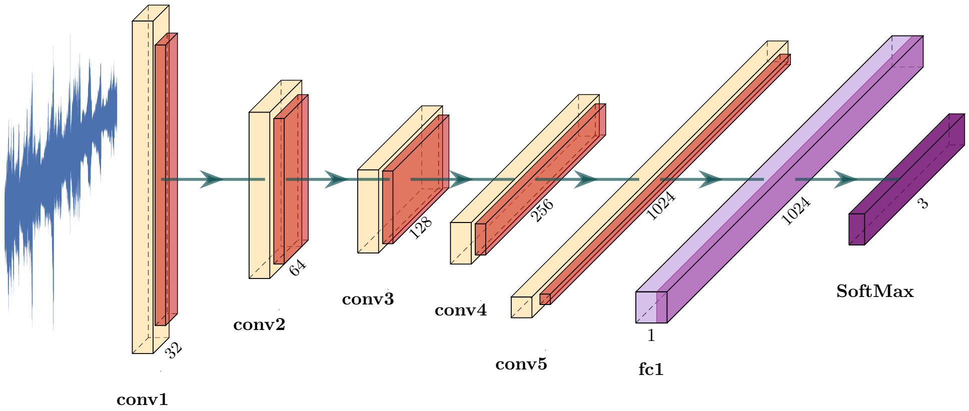


Figure 3.3. Our audio classifier architecture. After every convolutional layer we apply a 1D-Batch Normalization layer and 1D-MaxPolling, also we use LeakyRelu as activation function.

Table 3.1. The architecture of our sound classifier. $MaxP$ is MaxPolling operation, $AvgP$ is Adaptative Average Polling operation, LR is the LeakyRelu activation function, $Soft$ is the SoftMax function, $BN1$ is a BatchNormalization1D layer, $Conv1D$ is a one-dimensional standard convolutional layer, and $Fc1$ is Fully Connected layer. The $\circ$ operator defines the sequence of layers in our architecture.

Block	Operations	Tensor Size
1	$MaxP \circ LR \circ BN1 \circ Conv1D$	(1,64000)
2	$MP \circ LR \circ BN1 \circ Conv1D$	(32,8000)
3	$MP \circ LR \circ BN1 \circ Conv1D$	(64,1000)
4	$LR \circ BN1 \circ Conv1D$	(128,125)
5	$LR \circ BN1 \circ Conv1D$	(256,63)
6	$Soft \circ Fc1 \circ AvgP$	(1024,5)
7	Output	(1,3)

3.2 Latent Space Encoding for Motion Generation

In order to create movements that follow the music style, while keeping unique motion particularities and being temporally coherent, we build a latent vector that combines the extracted music style with a spatiotemporal correlated signal from a Gaussian process. It is noteworthy that our latent vector differs from the work of Yan et al. [2019], since we condition our latent space using the information provided by the audio classification. The information used to condition the motion generation and to create our latent space is a trainable dense representation of each class. The dense representation works as a categorical dictionary, which maps a dance style class to a higher dimensional space.

Generative models, more specifically GAN's, aim to generate samples from the real data distribution using a random noise, often a Gaussian noise. Gaussian process is a suitable mechanism to deal with temporal constraints and a collection of random variables that has a multivariate normal distribution. In this context, Gaussian process allows sampling from a distribution of functions, where there is a correlation between the points sampled for each function. In Figure 3.4, we vary the value of σ_c parameter of the Gaussian process shown in Equation 3.2, this variation in the signal frequency enables our model to infer which skeleton joint is responsible for more prolonged movements and explore a large variety of poses.

The temporal coherent signals are sampled from Radial Basis Function kernel (RBF) proposed by Rasmussen [2003] to enforce temporal relationship among the N frames. A zero-mean Gaussian process with a covariance function κ is given by $(z_t^{(c)}) \sim GP(0, \kappa)$, where $(z_t^{(c)})$ is the c-th component of z_t. The signal is composed of c functions,

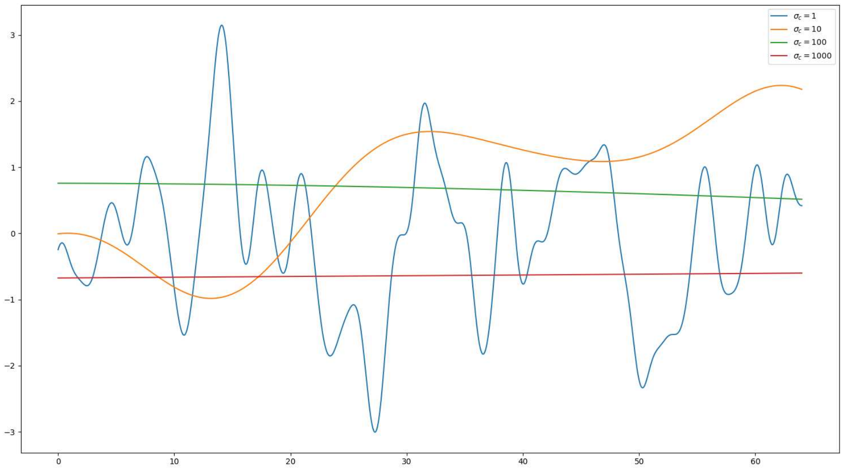


Figure 3.4. Example of the Gaussian process used to generate random noise. Four functions are sampled with different σ_c (frequency) values.

each function with $t, t' \in \mathbb{R}^{N/16}$ temporally coherent values. This process provides a signal with a shape of (C, T, V), where C is interpreted as the channels (features) of our graph, T is related to the length of the sequence we want to generate, and V is the spatial dimension of our graph signal. The covariance function κ is defined as:

$$\kappa(t, t') = \exp\left(-\frac{|t - t'|^2}{2\sigma_c^2}\right). \tag{3.2}$$

In our training experiments, we used $C = 512, T = 4, V = 1$ and $\sigma_c = \sigma\left(\frac{c_i}{C}\right)$, where $\sigma = 200$ was chosen empirically and c_i varies for every value from 1 to $C + 1$.

Then, we combine the temporal coherent random noise with the music style representation to generate coherent motions over time. Thus, the final latent vector is the result of concatenating the dense trainable representation of the audio class with the coherent temporal signal from the Gaussian process in the dimension of the features, as shown in Figure 3.5. This concatenation plays a key role in the capability of our method to generate synthetic motions with more than one dance style when the audio is a mix of different music styles. In other words, unlike a vanilla conditional generative model, which conditioning is limited to one class, we can condition over several classes over time.

The process to create our latent space is illustrated in Figure 3.5. In Figure 3.5 (a) we can sample from a space a set of functions given by the Gaussian process, thus in Figure 3.5 (a) we illustrate the sampling of three possible gaussian noise for our formulation. The classification of the input audio is represented by Figure 3.5 (b) where any classification algorithm receives the auditory data and returns a vector of probabilities of the input data belongs to each class, then we take this representation (vector of probabilities) and transform in a higher-dimensional space with dense representation, *i.e.*, the colors, this space is composed of trainable values (the colors will change during training stage). Finally, in Figure 3.5 (c) we compose our latent space by concatenating both information the gaussian noise and the dense representation of the music class. Our latent space, which will be fed in our generator has both information, the temporal constraints addressed by the gaussian noise from the Gaussian process, and the dense representation of the auditory data, which is responsible for condition the motion generation.

The final tensor representing our latent vector has the size $(2C, T, V)$, where C and T has the same size of the temporal coherent signal. Note that the length of the final sequence is proportional to T used in the creation of the latent vector. The final motion, after propagated in our motion generator, will have $2^l T = N$ frames, where l is the number of temporal upsampling layers in the architecture. Therefore, we can generate samples for any FPS, and any length by changing the dimensions of the latent vector, more precisely changing the parameter T in the size of the final tensor. Moreover, as we conditioned by the channels dimension, it is possible to change the conditioning dance style over time.

As an example, to generate a sequence with two dance styles we create a tensor of size $(2C, 2T, V)$, where the first part of the tensor has $(2C, T, V)$ values to condition the motion generation in one style and the other part to condition in another style. Note that the temporal dimension follows the Gaussian process so we have correlated values in each function, what we change is the $C \times T$ values we use to condition the generator. In the end, we have a sequence of $32T$ frames where in the first $16T$ frames of the motion are from one style, and in the left $16T$ frames the motion performed is from another dance style.

Following the commonly used notation of GAN's, the Gaussian process generates our random noise z and the dense representation of the dance style is the variable used to condition our model y. The combination of both data is used as input for the generator.

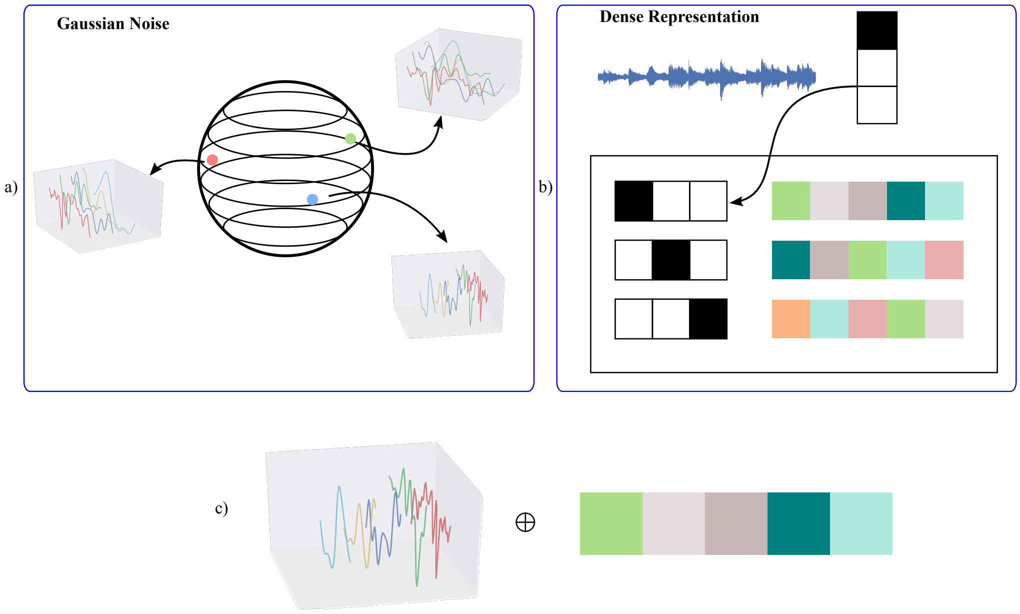


Figure 3.5. Illustration of our latent space. We first sample a set of functions using a Gaussian process, then we define the class of an input audio data, and transform the vector of probabilities of the classification in a higher dense representation of the most probable class, finally we concatenate both, the gaussian noise and the dense representation to create our latent space. (a) Samples of the Gaussian noise using the Gaussian process. (b) The dense representation used to conditioned the motion generation. (c) Our latent space. In figure (a) we sample three gaussian noise using the Gaussian process, where each function with different values for σ_c has its own color, for visualization purposes we illustrate the sampling space as a sphere. In (b) we illustrate the process of audio classification where a vector of probabilities is the output, for instance, in the figure the classify the audio as for the first class (black value in the vector), we then change the representation to a dense representation in a higher dimensional space, the colors in our illustration represent the values of the dense representation, which can change during training. In figure (c) we combine both the dense representation of the audio most probable class and the gaussina noise from the Gaussian process, concatenating both tensors in the channels dimension, the concatenation operator is defined as $\bigoplus$.

3.3 Upsampling & Downsampling Operators

When using GCNs, one challenge that appears in an adversarial training is the requirement of upsampling the latent vector in the spatial and temporal dimensions to fit the motion space $\mathcal{M}$ (Equation 3.1). To allow us to transform the latent space into a sequence of human poses, we need an upsampling operator. In practice since our latent space is organized in features, time, and vertex dimensions, we need two

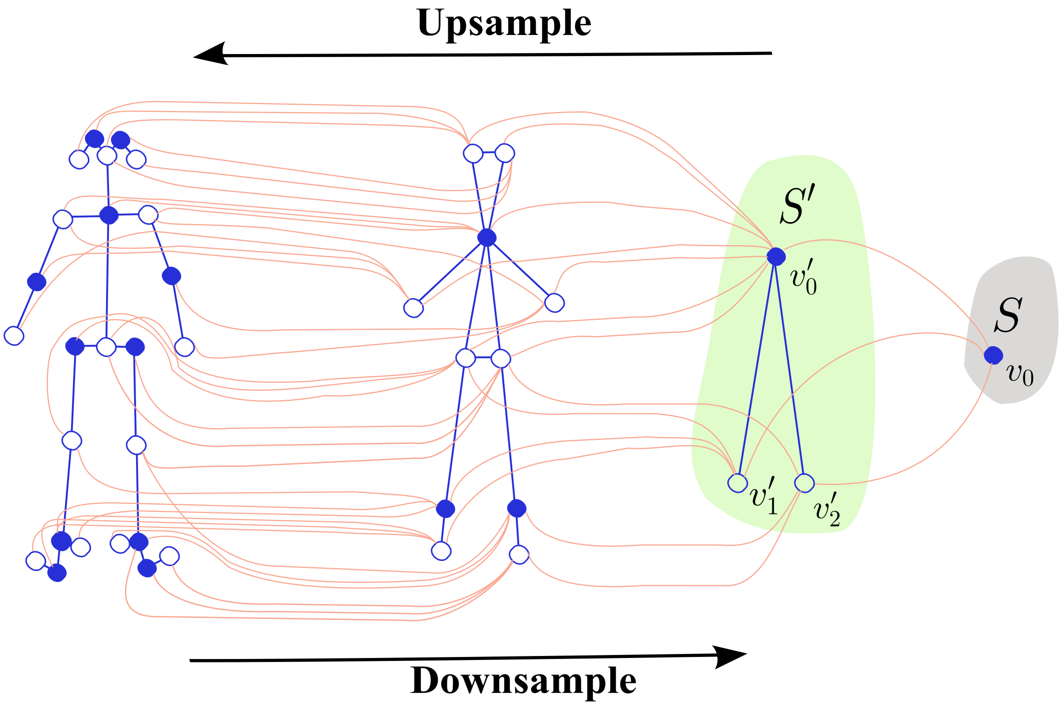


Figure 3.6. Graph schemes for upsampling and downsampling operations. The red links shows the relationship between vertex over each pair of graphs, and from right to left the graph S to the graph S' illustrates our first spatial upsampling operator.

upsampling operators, one of them for the time dimension and another for the vertex dimension. Our operators are based on the work presented by Yan et al. [2019], where an upsampling and downsampling operators are defined. The downsampling operator is useful for the discriminator network since the discriminator aims to reduce the dimensionality of the sequence of human poses to a probability of that sequence to be real or fake.

The first step to produce our upsampling and downsampling operators is to create an adjacency matrix that links smaller graphs to bigger ones, as shown in Figure 3.6. For example, in Figure 3.6 going for graph S (in blue) to the graph S' (in green), we create an adjacency matrix that follows the Algorithm 1.

With the adjacency matrix between every pair of graphs represented in Figure 3.6 created following the Algorithm 1 we can define our aggregation function, which will transform a smaller graph into a bigger one, and vice-versa.

Inspired by the work of Yan et al. [2019], we included in our architecture a spatial upsampling layer. This layer operates using an aggregation function defined by an adjacency matrix A^{ω} that maps a graph $S(V, E)$ (representing an human skeleton) with V vertices and E edges to bigger graph $S'(V', E')$ (see Figure 3.6). Differently

Algorithm 1: Algorithm to construct the adjacency matrix used by the upsampling and downsampling operators

```
Result: return adj
$S$;                                              // smaller graph
$S'$;                                              // bigger graph
$k = 2$;                      // geodesic distance used in our experiments
$adj$;                              // intialization of adjacency matrix
; $i = 0$;                 // variable to iterate over the geodesic distance
; while $i < k$ do
    foreach $v \in S$ do
        foreach $v' \in S'$ do
            if $geodesic\_distance(v, v') == i$ then
                $adj_{i,v,v'} = 1$;
            else
                $adj_{i,v,v'} = 0$;
            end
        end
    end
    $i + +$;
end
```

from Yan et al. [2019], our upsampling and downsampling operators can learn the best values of A^{ω} that leads to a good upsampling of the graph by assigning different importance of each neighbor to the new set of vertices. However, the matrix A^{ω} is initialized using the adjacency matrix result of the Algorithm 1, and the values change during the training stage. This modification we made in the operator defined by Yan et al. [2019] show that the method achieve the same results faster, than without it.

The first spatial upsampling layer starts from a graph with one vertex and then increases it to a graph with three vertices. When creating the new vertices, the features $\mathbf{f_j}$ from the initial graph S are aggregated by A^{ω} as follows:

$$\mathbf{f'}_i = \sum_{k,j} A^{\omega}_{kij} \mathbf{f}_j, \tag{3.3}$$

where $\mathbf{f'_i}$ contains the features of the vertices in the new graph S', and k indicates the geodesic distance between vertex j and vertex i in the graph S'.

For instance, in Figure 3.6, from right to left, we can see the upsampling operation, where we move from a graph with one vertex S to a new graph containing three vertices S'. The aggregation function in A^{ω} is represented by the red links connecting the vertices between the graphs and the topology of graph S'. When $k = 0$, vertex v is directly mapped to vertex v' (*i.e.*, the geodesic distance between v_0 and v'_0 is 0) and all values are zeros except the value of $i = 0, j = 0$ then $\mathbf{f'}_0 = A^{\omega}_{0,0,0}\mathbf{f}_0$. Following the example, when $k = 1$, we have $\mathbf{f'}_1 = A^{\omega}_{1,1,0}\mathbf{f}_0$ and $\mathbf{f'}_2 = A^{\omega}_{1,2,0}\mathbf{f}_0$. All the others values of A^{ω} are zeros.

The spatial downsampling layers follows the same procedure of upsampling operations but using an aggregation matrix B^{ϕ} with trainable weights ϕ, different from the weights learned by the generator. Again the matrix B^{ϕ} is initialized with the adjacency matrix result fo the Algorithm 1. Since the aggregation is performed from a large graph G' to a smaller one G, the final aggregation is given by

$$\mathbf{f}_i = \sum_{k,j} B^{\phi}_{kij}\mathbf{f'}_j. \tag{3.4}$$

Finally, the temporal upsampling and downsampling operators are implemented using traditional two-dimensional convolutions, in the case of the upsampling we use transposed convolutions, and for the downsampling operator classic convolutions, the same used in traditional CNN's models.

3.4 Spatio-Temporal Convolutional Operator

To deal with the temporal issue of the problem, we adopt the Spatial-Temporal Graph Convolutional Network (ST-GCN) proposed by Yan et al. [2018]. The layers used in these networks consider a sequence of graphs, more precisely, a set of human poses. The network aims to relate the temporal features of the subsequent graphs, and use this relationship to help in an action recognition task. The features of the graph can be 2D image joints' coordinates, as in our work, or 3D coordinates.

The layer defined by Yan et al. [2018] can be used as our spatio temporal operator since they use a sequence of graphs to infer actions an actor is doing over time. Their formulation follows the same principle of the formulation of traditional graph convolutional networks as proposed by Kipf and Welling [2017]. The aggregation function used in ST-GCN is defined as follows:

$$\mathbf{f_{out}} = \sum_{j} \Lambda_j^{-\frac{1}{2}}\mathbf{A_j}\Lambda_j^{-\frac{1}{2}}\mathbf{f_{in}}\mathbf{W_j}, \tag{3.5}$$

where $\mathbf{A_j}$ is the result of an element-wise product between the adjacency matrix with self-loops in every node and a matrix of learnable weights, and this is the main difference between their approach and the approach proposed by Kipf and Welling [2017] and defined in Equation 2.2, Λ denotes the diagonal node degree matrix, and $\mathbf{W_j}$ is a

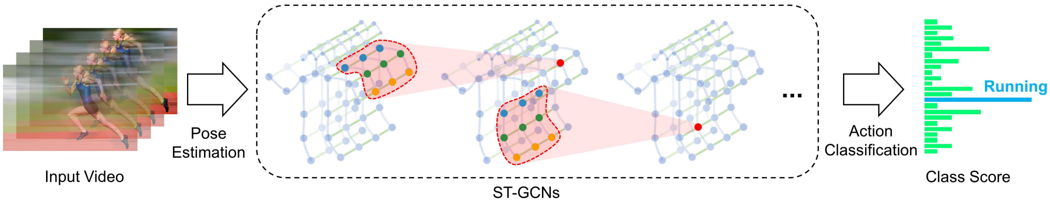


Figure 3.7. Spatial Temporal Graph Convolutional Networks applied in the task of action recognition, as proposed by Yan et al. [2018]. Image courtesy of Yan et al. [2018].

weight matrix for the j-th neural network layer. Moreover, the temporal issue of the problem is implemented using classic two-dimensional convolutions operators.

In practice to implement the graph operator of the ST-GCN layer, we use standards two-dimensional convolutions multiplied by the normalized adjacency matrix with the learnable weights. An illustration of the ST-GCN can be seen in Figure 3.7, where the first part of the image shows the input of the method, in that case, a video sequence. The center of Figure 3.7 illustrates the operations in the graphs over time, which is the ability we are interested, and the final part of Figure 3.7 shows the results of a classification task, since the problem addressed by Yan et al. [2018] was of action recognition.

3.5 Conditional Adversarial GCN for Motion Synthesis

To generate realistic movements, we trained our graph convolutional neural network (GCN) with an adversarial strategy. The key idea in adversarial conditional training is to learn the data distribution while two networks compete against each other in a *minimax* game. In our case, the motion generator G seeks to create new motion samples as similar as possible to those in the motion training set, while the motion discriminator D tries to distinguish generated motion samples (fake) from real motions of the training dataset (real). This training scheme is illustrated in Figure 3.8.

3.5.1 Generator

The architecture of our generator G is mainly composed of three types of layers: temporal and spatial upsampling operations, and spatial-temporal graph convolutions. The architecture of our generator is described in Table 3.2.

The temporal upsampling layer of our generator consists of transposed 2D convolutions that double the time dimension, this convolutions only change the shape of

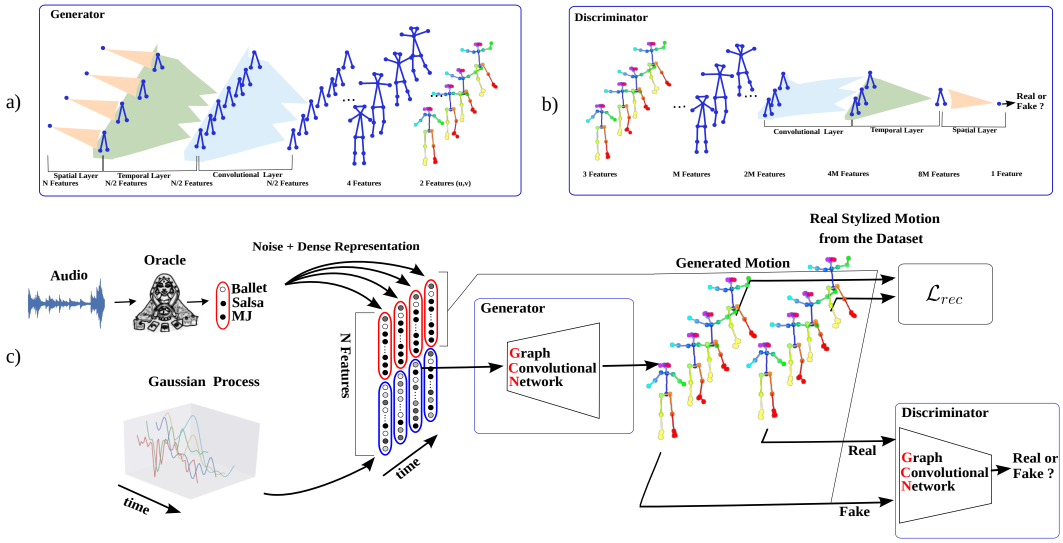


Figure 3.8. Overview of our method. (a) Our motion GCN Generator G; (b) Motion GCN Discriminator D; and (c) an overview of the adversarial training regime. Our motion generator with the upsampling operators transforms a one-dimensional latent space into a sequence of human poses. Our discriminator takes a sequence of human poses and tries to distinguish real samples from the fake (generated by the generator) ones. An to train our motion generation approach, we use an oracle algorithm, to define the class of the input music, combine the classification of the oracle, with the gaussian noise from the Gaussian process, feed our generator network, then the output is used in the classification of the discriminator network.

the tensor in the temporal dimension, ignoring all others dimensions. The spatial upsampling layer is defined in Section 3.3. The spatial-temporal graph convolutions are described in Section 3.4.

In the first layer of the generator, we have one node containing a total of N features; these features represent our latent space (half from the Gaussian Process and a half from the audio representation). The features of the subsequent layers are computed by the operations of upsampling and aggregation. The last layer outputs a graph with 25 nodes containing the (x, y) coordinates of each skeleton joint.

After applying the temporal and spatial upsampling operations, our generator uses the graph convolutional layers defined in Section 3.4. These layers are responsible to create the spatial-temporal relationship between the graphs. Then, the final architecture comprises three sets of temporal, spatial, and convolutional layers: first temporal upsampling for a graph with one vertex followed by an upsampling from one vertex to 3 vertices, then one convolutional graph operation. We repeat these three operations for the upsampling from 3 vertices to 11, and finally from 11 to 25 vertices, which represents the final pose. Figure 3.8-(a) illustrates this GCN architecture.

Table 3.2. The architecture of our motion generator. LR is the LeakyRelu activation function, $BN2$ is a BatchNormalization2D layer, $Drop$ is a standard dropout layer, GCN_{st} is a ST-GCN layer as proposed by Yan et al. [2018] and described in Section 3.4, Up_s is graph upsampling operator as defined in Section 3.3, and Up_t is a standard two-dimensional convolution with stride to always double the length of the sequence. The $\circ$ operator defines the sequence of layers in our architecture. The dimensions of the tensor are channels, time and vertex respectively.

Block	Operations	Input Tensor Size
1	$LR \circ GCN_{st}$	(1024,4,1)
2	$LR \circ BN2 \circ GCN_{st} \circ Up_s$	(512,5,1)
3	$Drop \circ LR \circ BN2 \circ GCN_{st} \circ Up_t$	(256,4,3)
4	$LR \circ BN2 \circ GCN_{st} \circ Up_s \circ Up_t$	(128,8,3)
5	$Drop \circ LR \circ BN2 \circ GCN_{st} \circ Up_t$	(64,16,11)
6	$LR \circ BN2 \circ GCN_{st} \circ Up_s \circ Up_t$	(32,32,11)
7	GCN_{st}	(16,64,25)
8	Output	(2,64,25)

Moreover, in all layers, we use LeakyRelu as an activation function, and after every upsampling followed by the graph convolutional layer we apply a Batch Normalization layer, furthermore, we apply dropout layers in every graph convolutional layer to avoid overfitting issues. The architecture of our generator is described in Table 3.2.

3.5.2 Discriminator

The discriminator D has the same architecture used by the generator but using downsampling layers instead of upsampling layers. Thus, all transposed 2D convolutions are converted to standard 2D convolutions, and the spatial downsampling layers follow the same procedure described in Section 3.3.

In the discriminator network, the feature vectors (the discriminator input) are assigned to each node as follows: the first layer contains a graph with 25 nodes, where their feature vectors are composed of the (x, y) coordinates on a normalized space and the class of the input motion. In the subsequent layers, the features of each node are computed by the operations of downsampling and aggregation. The last layer contains only one node that outputs the classification of the input data being fake or real. Figure 3.8-(b) illustrates the discriminator architecture. We use the same activation function as used in the Generator (LeakyRelu), and the same procedures with BatchNormalization and Dropout layers were adopted in the discriminator as well, however, the dropout layers in the discriminator are being applied together with

Table 3.3. The architecture of our motion discriminator. LR is the LeakyRelu activation function, Sig is the Sigmoid function, $BN2$ is a BatchNormalization2D layer, GCN_{st} is a ST-GCN layer as proposed by Yan et al. [2018] and described in Section 3.4, Dw_s is graph downsampling operator as defined in Section 3.3, and Dw_t is a standard two-dimensional convolution with stride to always cut half of the length of the sequence. The $\circ$ operator defines the sequence of layers in our architecture. The dimensions of the tensor are channels, time and vertex respectively.

Block	Operations	Input Tensor Size
1	$LR \circ GCN_{st}$	(3,64,25)
2	$LR \circ Dw_t \circ GCN_{st}$	(2,64,25)
3	$LR \circ BN2 \circ Dw_s \circ Dw_t \circ GCN_{st}$	(32,32,25)
4	$LR \circ BN2 \circ Dw_t \circ GCN_{st}$	(64,16,11)
5	$LR \circ BN2 \circ Dw_s \circ Dw_t \circ GCN_{st}$	(128,8,11)
6	$Dw_t \circ GCN_{st}$	(256,4,3)
7	Sig	(1,1,1)
7	Output	(1)

the spatial-temporal operator. The architecture of our discriminator is described in Table 3.3.

3.5.3 Adversarial Training

Given the motion generator and discriminator, our conditional adversarial network aims at minimizing the binary cross-entropy loss:

$$\begin{aligned}\mathcal{L}_{cGAN}(G, D) = \min_G \max_D \big(&\mathbb{E}_{x\sim p_{data}}(x)[\log D(x|y)]+\\ &\mathbb{E}_{z\sim p_z}(z)[\log(1 - D(G(z|y)))]\big)\,,\end{aligned} \tag{3.6}$$

where the generator aims to maximize the error of the discriminator, while the discriminator aims to minimize the classification fake-real error shown in Equation 3.6. In particular, in our problem, p_{data} represents the distribution of real motion samples, $x = \mathcal{M}_\tau$ is a real sample from p_{data}, and $\tau \in [0 - \mathcal{D}_{size}]$ and $\mathcal{D}_{size}$ is the number of real samples in the dataset. Figure 3.8-(c) shows a concise overview of the steps in our adversarial training.

The latent vector, which is used by the generator to synthesize the fake samples x', is represented by the variable z, the coherent temporal signal. The dense representation of the dance style is determined by y, and p_z is a distribution of all possible temporal coherent latent vectors generated by the Gaussian process. The data used by the generator G in the training stage is the pair of temporal coherent latent vector z, with a real motion sample x, and the value of y guaranteed by an oracle algorithm that returns the dance style to be used in the dense audio representation.

To improve the generated motion results, we designed a motion reconstruction loss term using L_1 distance in all skeletons over the N motion frames as follows:

$$\mathcal{L}_{rec} = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_{pose}(\mathbf{P}_t, \mathbf{P}'_t), \tag{3.7}$$

with $\mathbf{P}'_t \in \mathcal{M}'$ is the generated pose and $\mathbf{P}_t \in \mathcal{M}$ is a real pose from the training set and extracted with the OpenPose framework presented in the work of Cao et al. [2019]. The pose distance is computed as $\mathcal{L}_{pose} = \frac{1}{25}\sum_{i=0}^{24} |\mathbf{J}_i - \mathbf{J}'_i|_1$, following the notation shown in Equation 3.1.

Our final loss is then a weighted sum of the motion reconstruction and cGAN discriminator losses given by

$$\mathcal{L} = \mathcal{L}_{cGAN} + \lambda\mathcal{L}_{rec}, \tag{3.8}$$

where λ weights the reconstruction term. The λ value was chosen empirically, and was constant throughout the training stage. The initial guess regarding the magnitude of λ followed the values chosen in the work of Wang et al. [2018].

In the last step, when generating the motion, we apply a cubic-spline interpolation proposed by De Boor et al. [1978] to remove eventual high-frequency artifacts present in the generated motion frames $\mathcal{M}'$. Moreover, all motions generated by our formulation are in a normalized space to avoid space translation and scale problem during the trainnig. In order to handle different shapes of the actors and to reduce the effect of translations in the 2D poses of the joints, we normalized the motion data used during the adversarial GCN training. We managed changes beyond body shape and translations, such as the situations of actors lying on the floor or bending forward, by selecting the diagonal distance of the bounding box encapsulating all 2D body joints $\mathbf{P}_t$ of the frame as scaling factor. The normalized poses are given by:

$$\bar{\mathbf{J}}_i = \frac{1}{\delta}\left(\mathbf{J}_i - \left(\frac{\Delta u}{2}, \frac{\Delta v}{2}\right)\right) + 0.5, \tag{3.9}$$

where $\delta = \sqrt{(\Delta u)^2 + (\Delta v)^2}$, and $(\Delta u, \Delta v)$ are the differences between right-top position and left-bottom position of the bounding box of the skeleton in the image coordinates (u, v). This normalization removes problems of skeleton scale and translation in the image and puts the range of the joint poses for training $\bar{\mathbf{J}}_i \in [0, 1]$, without

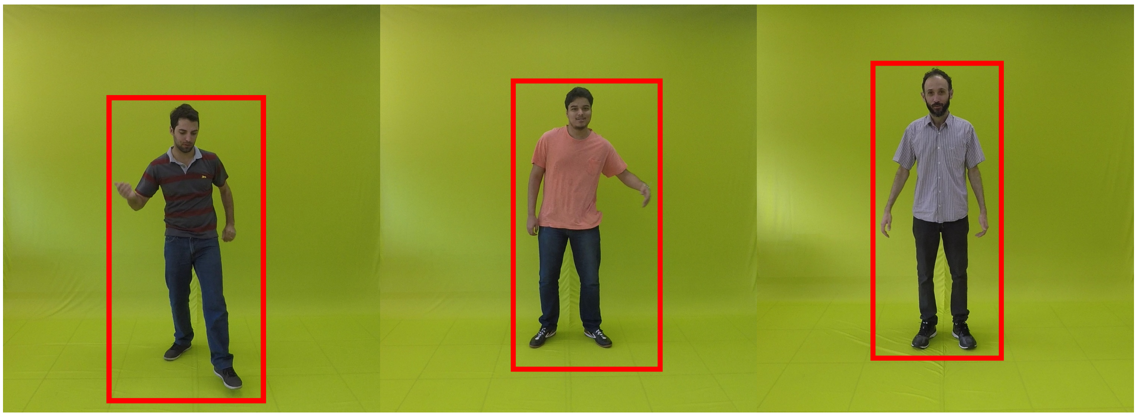


Figure 3.9. Illustration of the normalization process. We use the bound box in red to define the new space of 2D joints' coordinates, thus every joint in the new space has a normalized scale without necessarily being on the limits of the space. Translation issues are solved by this formulation.

necessarily having at least one joint **J** in the interval limits. An illustration of the bound box can be seen in Figure 3.9, the bound box varies according to the actors' poses, removing translation problems and reducing the scale issues.

3.6 Virtual Avatar Animation

The final step of our formulation, is to animate a virtual avatar using the generated motions to different musics, thus we can synthesize a video sequence of an actor performing a synthetic motion. The image-to-image translation technique vid2vid presented by Wang et al. [2018] was selected to synthesize videos from the avatar given sequences of poses generated by our method. We trained vid2vid to generate new images for these avatars, following the multi-resolution protocol described in Wang et al. [2018].

For inference, we feed the synthesized motion by our generator to the vid2vid framework, after desnormalizing the output following the inverse operation of Equation 3.9 and apply the splines described in Section 3.5.3. We highlight that any video translation style transfer method can be used with few adaptations, as for instance the works of Chan et al. [2019] and Gomes et al. [2020].

For visualization purposes we show in Figure 3.10 three actors used in our experiments. We also present the synthetic views generated by the trained vid2vid for each actor. The first column original images of the actors used in the training phase of vid2vid, the second column shows a synthetic pose generated by our approach after the denormalization process, and in the third column the virtual avatars for each actor performing the same synthetic pose generated by our method, after the denormalization using the inverse operation of Equation 3.9.

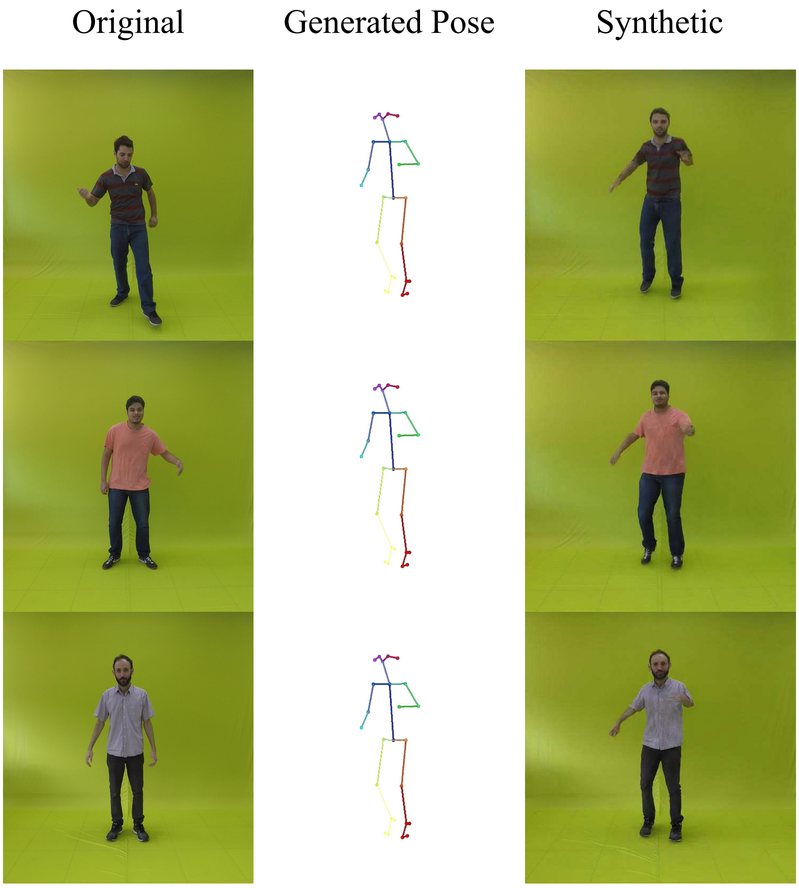


Figure 3.10. Actors, synthetic pose and the respective avatar animations. In the first column are real images from the actors, used to train vid2vid framework presented by Wang et al. [2018]. The second column, shows the synthetic pose generated by our graph convolutional motion generator. In the third column, we show the output of the vid2vid for the same synthetic human pose generated by or method, note that the actors in the synthetic image have a similar height, because of our normalization operation.

Chapter 4

Audio-Visual Dance Dataset

To train and evaluate the motion generation from music, we need a dataset with representatives movements for different music/dance styles. To our knowledge, only one publicly available dataset exists for the considered problem. However, their dataset is structured in dance units, small sets of consecutive frames to represent a typical movement for a dance style, the problem with dance units are that they are usually to small to represent a motion. Furthermore, the dataset presented by Lee et al. [2019] instead of prioritizing the quality of the data, the authors collected a huge amount of video data and create an algorithm to automatically detect representative movements in the videos. The difference in the structure of the dataset (dance units) and lower quality of poses, to favor the quantity of data, are two major drawbacks of their dataset.

Therefore build a new dataset composed of paired videos of people dancing three different music styles. The dataset is used to train and evaluate the methodologies for motion generation from audio. We split the samples into training and evaluation sets that contain multimodal data for the following music/dance styles: Ballet, Michael Jackson, and Salsa. These two sets are composed of two data types: *i)* visual data from carefully-selected parts of publicly available videos of dancers performing representative movements of the music style and, *ii)* audio data from the styles we are training. Figure 4.1 shows some data samples of our dataset. We also highlight two samples of the Ballet dance class for visualization purposes, shown in Figure 4.2.

The styles in our dataset were selected to be representative and challenging at the same time. By choosing Salsa, we selected a representative dance style with a plurality of movements performed by male and female dancers. With the Michael Jackson style, we attempted to explore the capability of synthesizing motions of a specific choreography: the movements of Michael Jackson are characteristic and also have a significant diversity of movements. Ballet style has the most characteristic

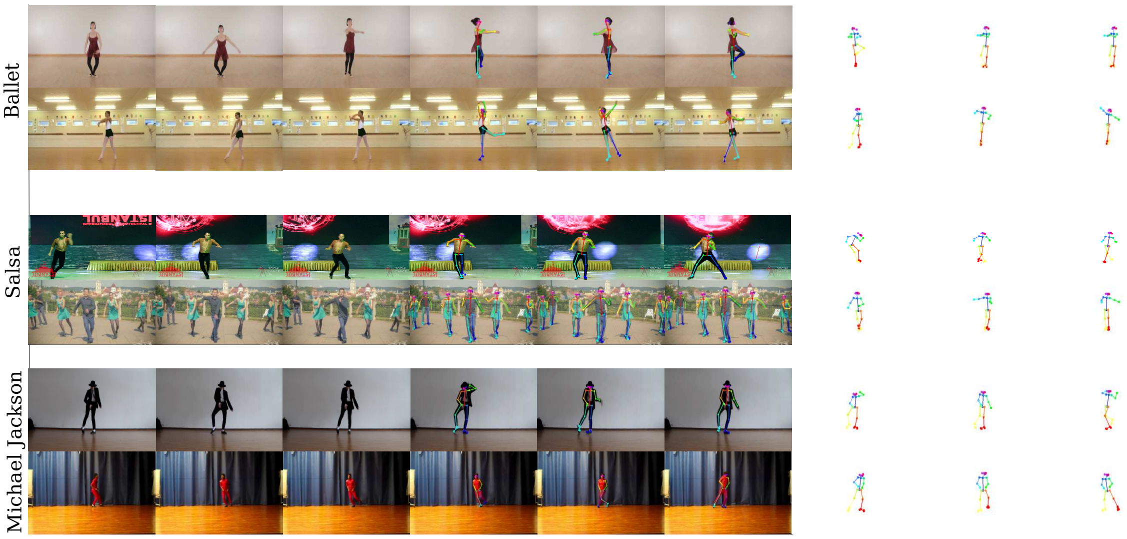


Figure 4.1. Video samples of the multimodal dataset with carefully annotated audio and 2D human motions of the three different dance styles.

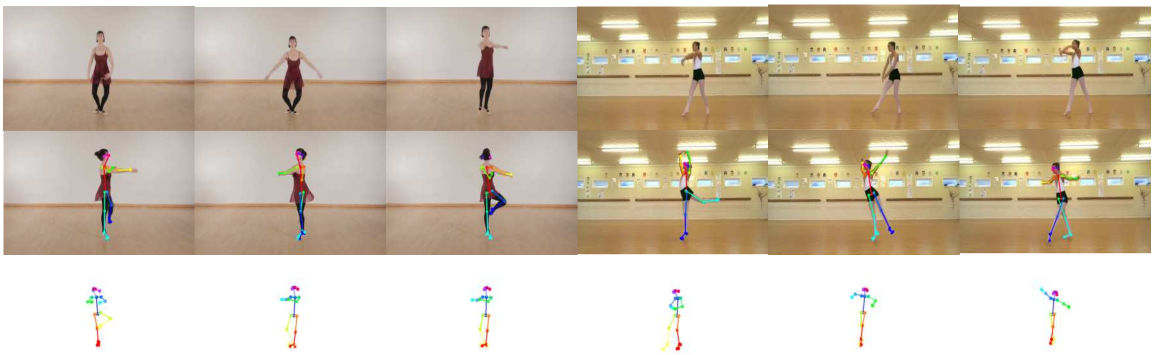


Figure 4.2. Example of two motion samples from the Ballet class on our dataset.

movements of the three styles, being challenging to the learning approach since the human poses are not conventional.

In order to collect meaningful audio information, several playlists from YouTube were chosen with the name of the style/singer as a search query. The audios were extracted from the resulting videos of the search and resampled to the standard audio frequency of 16KHz. For the visual data, we started by collecting videos that matched the music style and that had representative moves. Each video was manually cropped in parts of interest, by selecting representative moves for each dance style present in our dataset. Then, we standardize the motion rate throughout the dataset and convert all videos to 24 frames-per-second (FPS), maintaining a constant relationship between the number of frames and speed of movements of the actors. We annotate the 25 2D

Table 4.1. Dataset statistics for data used for the training and evaluation. The bold values are the number of samples used in the experiments.

	Training Dataset				Evaluation Dataset			
Setup	Ballet	MJ	Salsa	Total	Ballet	MJ	Salsa	Total
w/o Data Augmentation	16	26	27	69	73	30	126	229
w/ Data Augmentation	525	966	861	**2,352**	134	102	235	**471**

human joint poses for each video by estimating the pose with OpenPose presented by Cao et al. [2019]. Each motion sample is defined as a set of 2D human poses of 64 consecutive frames.

To improve the quality of the estimated poses in the dataset, we handled the miss-detection of joints by exploiting the body dynamics in the video. Since abrupt motions are not expected in the joints in a short interval of frames, we recreate a missing joint and apply the transformation chain of its parent joint. In other words, we infer the missing-joint position of a child's joint by making it follow its parent movement over time. Thus, we can keep frames with a miss-detected joint on our dataset.

4.1 Motion Augmentation

We also performed motion data augmentation to increase the variability and number of motion samples. We used the Gaussian process described in Section 3.2 to add temporally coherent noise in the joints lying in legs and arms over time. Also, we performed temporal shifts (strides) to create new motion samples.

For the training set, we collected 69 samples and applied the temporal coherent Gaussian noise and a temporal shift of size 32. In the evaluation set, we collected 229 samples and applied only the temporal shift of size 32 for Salsa and Ballet and 16 for Michael Jackson because of the lower number of samples (see Table 4.1). The temporal Gaussian noise was not applied in the evaluation set. The statistics of our dataset are shown in Table 4.1. The resulting audio-visual dataset contains thousands of coherent video, audio, and motion samples that represent characteristic movements for the considered dance styles.[1]

To evaluate the importance of the data augmentation in our results we performed evaluations with the same architecture and hyperparameters. We use the Fréchet Inception Distance (FID), a common metric for generative models, first introduced in the work of Heusel et al. [2017], the FID is calculated by estimating the distances between

[1] The dataset and project are publicly available at: https://www.verlab.dcc.ufmg.br/motion-analysis/cag2020.

two distributions of features vectors, one distribution from the generated data, and other from the real data, the closer the distance the better the results of the generative model. The results showed that without data augmentation, the performance on the Fréchet Inception Distance (FID) metric was on average 3 times worse than when using data augmentation. Moreover, we observed that the motions did not present variability, the dance styles were not well pictured, and in the worst cases, body movements were difficult to notice.

Chapter 5

Experiments and Results

To assess our method, we conduct several experiments evaluating different aspects of motion synthesis from audio information. We also compared our method to the state-of-the-art technique proposed by Lee et al. [2019], hereinafter referred to as D2M. We choose to compare our method to D2M since other methods does not allow a fair comparison without major modifications. As an example, the work proposed by Ginosar et al. [2019], presents two major drawbacks that make a comparison with our method unsuitable: *i)* Their work does not synthesize human motions for the entire human body, only for the upper half part of the human skeleton; *ii)* Their approach is actor specific, *i.e.*, it is not conditioned over classes and must be re-trained for every new person the method is trying to generate motions.

The experiments are as follows: *i)* We performed a perceptual user study using a blind evaluation with users trying to identify the dance style of the dance moves. Given a generated dance video, the user was asked to choose what style the avatar on the video is dancing (*e.g.*, Ballet, Michael Jackson (MJ), or Salsa); *ii)* Aside from the user study, we also evaluated our approach on commonly adopted quantitative metrics in the evaluation of generative models, such as Frechet Inception Distance (FID) proposed by Heusel et al. [2017], GAN-train, and GAN-test both proposed by Shmelkov et al. [2018].

5.1 Implementation and Training Details

In this section, we present the implementation details, and the training details for both, the GCN and the 1D-CNN audio classifier. Moreover, we provide details on the implementation of our competitor, since during our experiments their method was not fully publicly available.

5.1.1 Audio Classifier and Competitor Details

The one-dimensional audio CNN presented in Section 3.1 was trained for 500 epochs, with batch size equal to 8, Adam optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$, and learning rate of 0.01. Similar to work of van den Oord et al. [2016], we preprocessed the input music audio using a $\mu - law$ non-linear transformation to reduce noise from audio inputs that were not recorded properly.

To find the best hyperparameters, we run a 10-fold cross-validation and kept the best model to predict the music style to condition the generator. We highlight that our architecture is one-dimensional and works directly in the waveform, conversely to the works of Arandjelovic and Zisserman [2018] and Hershey et al. [2017] that require 2D pre-processed sound spectrograms, which show lower performances during our experiments. Moreover, during our experiments, the classifier shows always an average accuracy above 85% in the validation set.

To evaluate the capability of our classifier in extracting good features for auditory data, we apply a dimensionality reduction method. More precisely the T-distributed Stochastic Neighbor Embedding (t-SNE) method presented by Maaten and Hinton [2008]. The visualization of the feature space of our one-dimensional classifier can be seen in Figure 5.1, and show that our one-dimensional audio classifier can extract good features for the auditory data. We can notice the harder music style for the classifier was Salsa, as we can see in Figure 5.1, the Salsa class was the class with more misclassification instances.

Unfortunately, due to the lack of some components in the publicly available implementation of D2M, few adjustments were required in their audio preprocessing step. We standardized the input audio data by selecting the maximum length of the audio divisible by 28, defined as L, and reshaping it to a tensor of dimensions $\left(\frac{L}{28}, 28\right)$ to match the input dimensions of their architecture.

5.1.2 Training

We trained our GCN adversarial model for 500 epochs. We observed that additional epochs only produced slight improvements in the resulting motions. In our experiments, we select $N = 64$ frames, roughly corresponding to a motion between two to three seconds at 24 frames-per-second. We select 64 frames as the size of our samples to follow a similar setup presented in the work of Ginosar et al. [2019]. Moreover, the motion sample size in the work of Lee et al. [2019] also adopted motion samples of 32 frames. In general, the motion sample size is a power of 2, because the nature of the conventional convolutional layers adopted in both Ginosar et al. [2019] and Lee

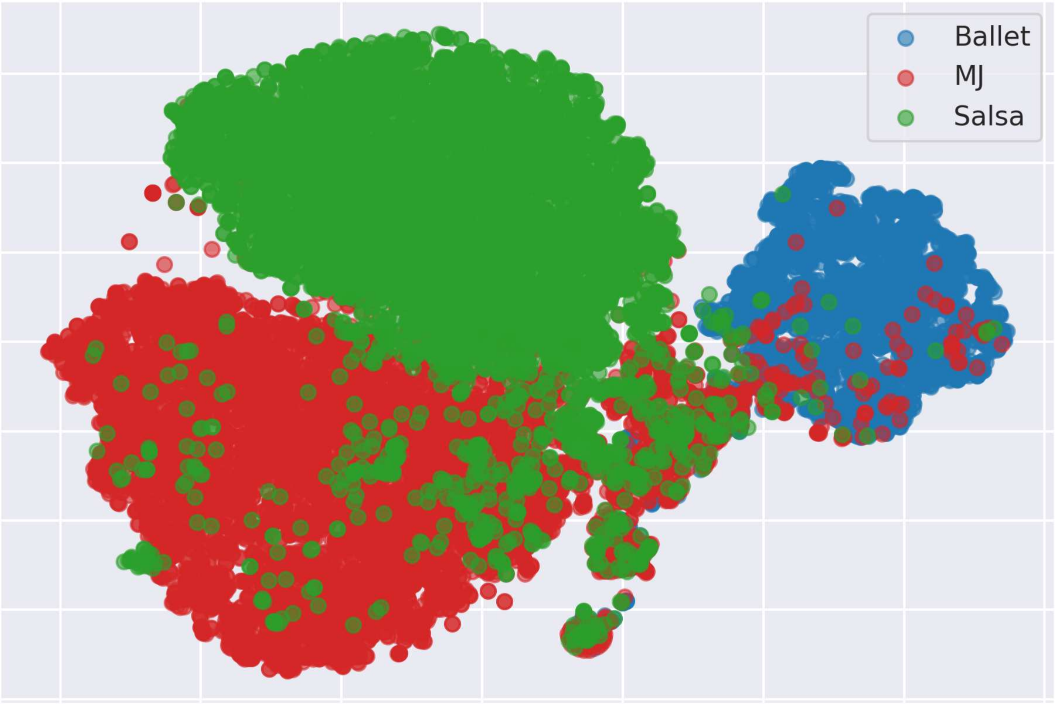


Figure 5.1. Result of the t-SNE algorithm over our 1D-CNN audio classifier model, we show approximately 18k samples. In red samples of Michael Jackson class, blue samples of Ballet class, and in green samples of Salsa class.

et al. [2019]. However, it is worth noting that our method is able to synthesize long motion sequences. We use a batch size of 8 motion sets of N frames each. We optimize our graph convolutional conditional adversarial network with Adam optimizer for the generator with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The discriminator was optimized with Stochastic Gradient Descent (SGD) using $\lambda = 100$.

We also train the vid2vid method proposed by Wang et al. [2018] not following the standard training protocol. The standard training protocol of vid2vid, our video transfer technique, uses data from OpenPose framework proposed by Cao et al. [2019] and DensePose framework proposed by Güler et al. [2018], the information provided by DensePose framework is a 2.5D representation of the human pose. Since our method cannot provide data in a 2.5D representation of the human pose, we modify the training protocol of the vid2vid framework, thus making possible the last step of our formulation, the virtual avatar animation. Note that, with this modification vid2vid framework only uses OpenPose data to train, the same kind of data we can generated using our motion generator.

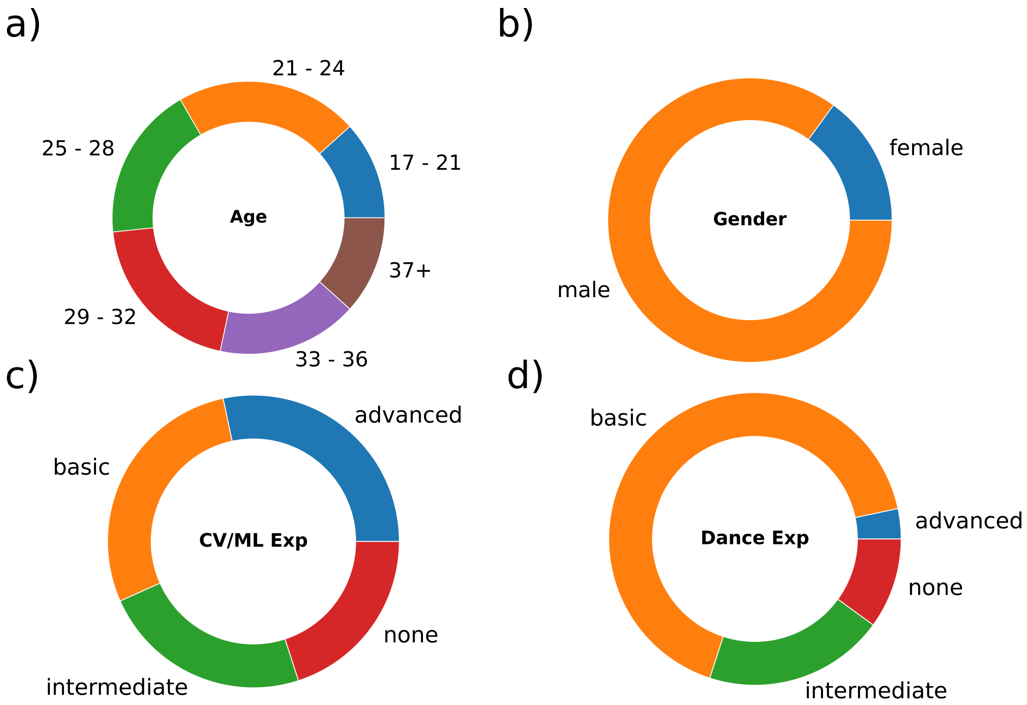


Figure 5.2. Profile of the participants of our User Study. The plots a), b), c) and d) show the profile distribution of the participants of our user study. The participant profile is defined as his/her age, gender, experience with computer vision and machine learning, familiarity with different dance styles.

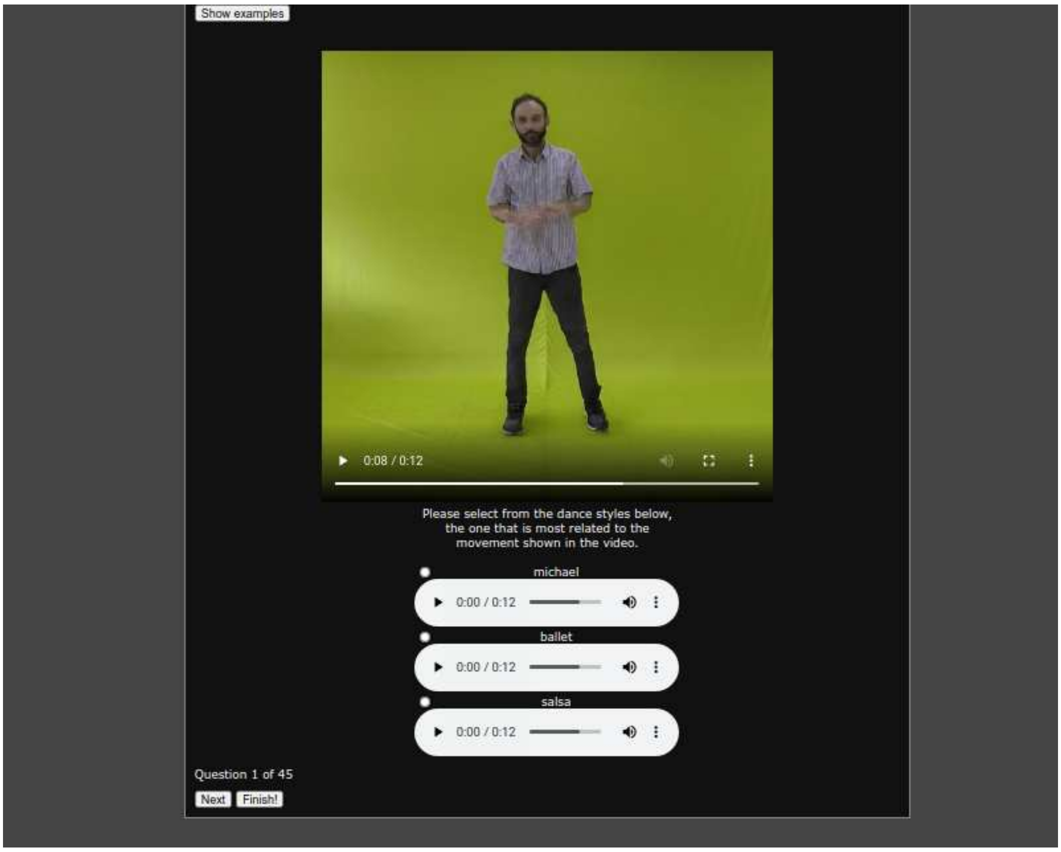
5.2 User Study

We conducted a perceptual study with 60 users and collected the age, gender, computer vision/machine learning experience, and familiarity with different dance styles for each user. Figure 5.2 shows the profiles of the participants. Figure 5.4 show the results of the user study, that will be discussed further in this section.

The perceptual study was composed of 45 randomly sorted videos. For each video, the user watches the video (with no sound) synthesized with the vid2vid framework from generated 2D poses and then was asked to associate the motion performed on the synthesized video as belonging to one of the audio classes (*i.e.*, Ballet, Michael Jackson, or Salsa). In each video, the users were supposed to listen to three audios (one for each audio class) to help them to classify the video. We had 60 users participating in our study. The website used to conduce the user study publicly available[1]

The set of videos was composed of 15 videos of movements generated by our approach, 15 videos generated by D2M proposed by Lee et al. [2019], and 15 placebos,

[1] https://www.verlab.dcc.ufmg.br/rhythm2motion



Please select from the dance styles below, the one that is most related to the movement shown in the video.

Question 1 of 45

Figure 5.3. User interface used in the user study. The participants watch a video and listen to three audios, then they must choose which dance class the motion performed in the video sequence is more related to. The participants could stop answering questions any time, and they could also re-watch some examples of real artists performing motions of the dance styles we were evaluating any time.

i.e., videos of real movements extracted from our training dataset. We applied the same transformations to all data, thus every video should have an avatar performing a motion with a skeleton with approximately the same dimensions. Also, we split the 15 videos shown by each one of the evaluated methods equally in three dance styles. Thus, we have 5 videos of each dance class, for each method in our study.

From Table 5.1 and Figure 5.4, we draw the following observations: first, our method achieved similar motion perceptual performance to the one obtained from real data. Second, our method outperformed the D2M method with a large margin. Thus, we argue that our method was capable of generating realistic samples of movement taking into account two the following aspects: *i)* Our results are similar to the real data results in a blind study; *ii)* Users show higher accuracy in categorizing our generated motion. Furthermore, as far as the quality of a movement being individual is concerned, Figures 5.5, 5.6, 5.7 and 5.8 show that our method was also able to generate samples

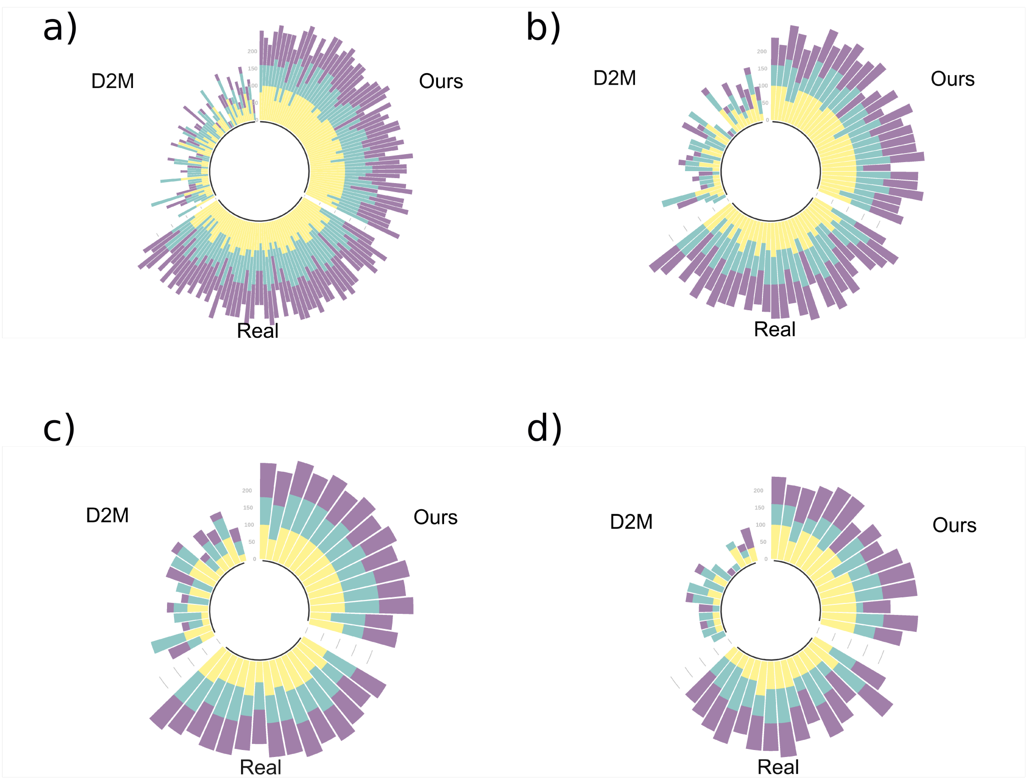


Figure 5.4. The plots a), b), c) and d) show the results of the study. In the plots of semi-circles are shown the results of the user evaluation; each stacked bar represents one user evaluation and the colors of each stacked bar indicates the dance styles (Ballet = yellow, Michael Jackson (MJ) = blue, and Salsa = purple). **a)** We show the results for all **60** users that fully answered our study; **b)** Results for the users which achieved top **27**% scores and the **27**% which achieved the bottom scores; **c)** Results for the **27**% user which achieved top scores; **d)** Results for the **27**% users which achieved worst scores.

with motion variability among samples.

In order to test the validity of the questions in the study, we ran two statistical tests used in item analysis: Difficulty Index and Item Discrimination Index. The Difficulty Index measures how easy to answer an item is by determining the proportion of users who answered the question correctly, *i.e.*, the accuracy. On the other hand, the Item Discrimination Index measures how a given test question can differentiate between users that mastered the motion identification from those who have not. Our methodology analysis was based on the guidelines described by Luger and Bowles [2016]. The average values of the indexes for all questions in the study are shown in the Table 5.1. One can clearly observe that our method's questions had a higher difficulty index value, which means it was easier for the participants to answer them correctly

Table 5.1. Quantitative metrics for user perceptual study

	Difficulty Index[1]			Discrimination Index [2]		
Dance Style	D2M	Ours	Real	D2M	Ours	Real
Ballet	0.183	0.943	**0.987**	**0.080**	**0.080**	0.033
MJ	0.403	0.760	**0.843**	0.140	**0.240**	0.120
Salsa	0.286	**0.940**	0.820	0.100	0.030	**0.180**
Average	0.290	0.881	**0.883**	0.106	**0.116**	0.111

[1] *Better closer to 1.*

[2] *Better closer to 1.*

and, in some cases, even easier than the real motion data. Regarding the discrimination index, we point out that the questions cannot be considered good enough to separate the ability level of those who took the test, since items with discrimination indexes values between 0 and 0.29 are not considered good selectors as addressed by Ebel and Frisbie [1991]. These results suggest that our method and the videos obtained from real sequences look too natural for most users, while the videos generated by the method proposed by Lee et al. [2019] were confusing.

5.3 Quantitative Evaluation Metrics

For a more detailed performance assessment regarding the similarity between the learned distributions and the real ones, we use the commonly used Fréchet Inception Distance (FID), as explained in Chapter 4 the Fréchet Inception Distance (FID), is a common metric for generative models, first introduced in the work of Heusel et al. [2017], the FID is calculated by estimating the distances between two distributions of features vectors, one distribution from the generated data, and other from the real data, the closer the distance the better the results of the generative model. We computed the FID values using motion features extracted from the action recognition Spatial-Temporal Graph Convolutional Network (ST-GCN) model presented by Yan et al. [2018], similar to the metric used in the works of Yan et al. [2019]; Lee et al. [2019]. We train the ST-GCN model 50 times using the same set of hyperparameters, since there is no such established action recognition method for extracting reasonable features vectors. The trained models achieved high accuracy scores, above 90% for almost all 50 training trials. It is noteworthy that the data used to train the feature vector extractor was not used to train any of the methods evaluated in this thesis. The results for FID metric are shown in Table 5.3.

We also computed two other well-known GAN evaluation metrics proposed

Table 5.2. Quantitative values of FID for generative models. We use as feauture extractor the work of Yan et al. [2018], which as been designed to address the problem of action recognition.

	FID[1]		
Dance Style	D2M	Ours	Real
Ballet	20.20 ± 4.41	**3.18 ± 1.43**	2.09 ± 0.58
MJ	**4.38 ± 1.94**	8.03 ± 3.55	5.60 ± 1.42
Salsa	12.23 ± 3.20	**4.29 ± 2.38**	2.40 ± 0.75
Average	12.27 ± 7.27	**5.17 ± 3.33**	3.36 ± 1.86

[1] *Better closer to 0.*

Table 5.3. Quantitative evaluation metrics: GAN-Train and GAN-Test.

	GAN-Train [1]			GAN-Test [2]		
Dance Style	D2M	Ours	Real	D2M	Ours	Real
Ballet	0.36 ± 0.15	**0.89 ± 0.10**	0.80 ± 0.12	0.07 ± 0.04	**0.80 ± 0.14**	0.77 ± 0.11
MJ	0.34 ± 0.15	**0.60 ± 0.13**	0.59 ± 0.04	**0.70 ± 0.14**	0.46 ± 0.18	0.60 ± 0.09
Salsa	**0.32 ± 0.17**	0.31 ± 0.11	0.50 ± 0.16	0.26 ± 0.14	**0.96 ± 0.11**	0.90 ± 0.06
Average	0.34 ± 0.16	**0.60 ± 0.26**	0.63 ± 0.17	0.34 ± 0.29	**0.74 ± 0.25**	0.76 ± 0.15

[1] *Better closer to 1.* [2] *Better closer to 1.*

by Shmelkov et al. [2018]: GAN-Train and GAN-Test. To compute the values of the GAN-Train metric, we trained the ST-GCN presented by Yan et al. [2018] in a set composed of dancing samples generated by our method and another set with generated motions by D2M. Then we tested the model in the evaluation set (real samples). The GAN-Test values were obtained by training the same classifier in the evaluation set and tested in the sets of generated motions. For each metric, we ran 50 training rounds and reported the average accuracy with the standard deviation in Table 5.3.

We can also note that the generator performs better in some dance styles. Since some motions are more complicated than others, the performance of our generator can be better synthesizing less complicated motions related to a particular audio class related to a dance style. For instance, the Michael Jackson style contains a richer set of motions with the skeleton joints rotating and translating in a variety of configurations. The Ballet style, on the other hand, is composed of fewer poses and consequently, easier to synthesize.

5.4 Qualitative Visual Results

We show some qualitative results in Figures 5.5, 5.6, 5.7, 5.8 and 5.9. We can notice that the sequences generated by D2M presented some characteristics clearly inherent

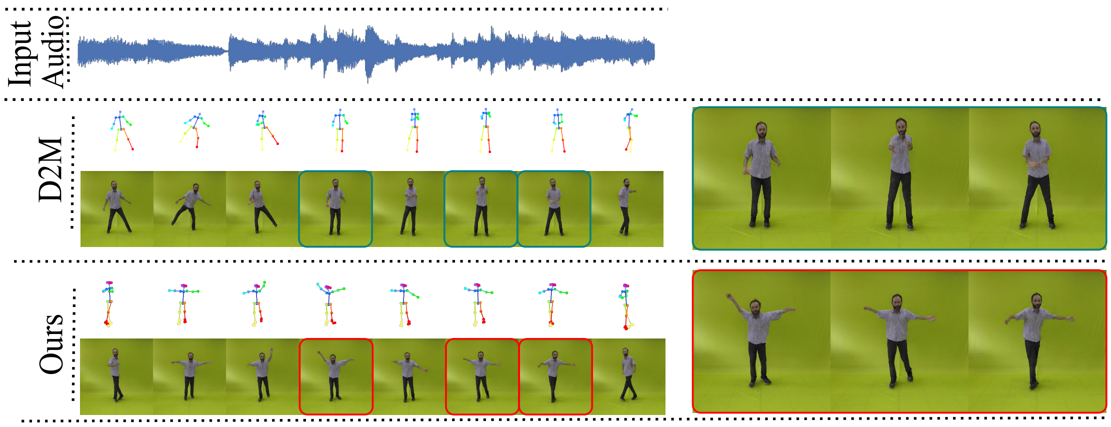


Figure 5.5. Results of our approach in comparison to D2M presented by Lee et al. [2019] for *Ballet*, the dance style shared by both methods. We highlight some frames generated by both method, showing that our method holds more characteristics of the original motion in the synthetic one.

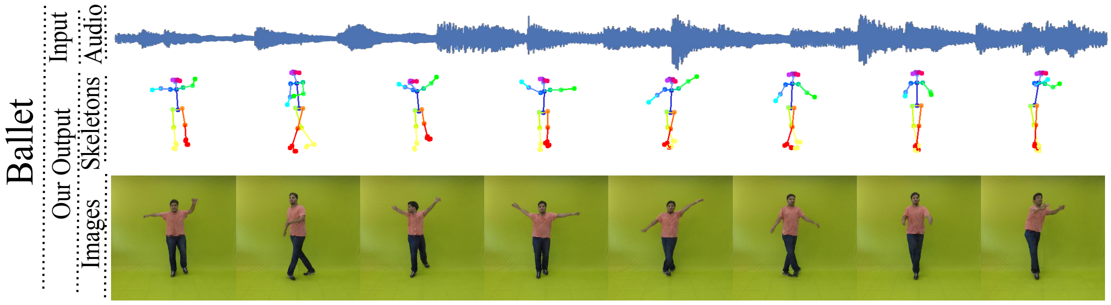


Figure 5.6. Qualitative results using audio sequences for Ballet dance style. First row: input audio; Second row: the sequence of skeletons generated with our method; Third row: the animation of an avatar by vid2vid using our skeletons.

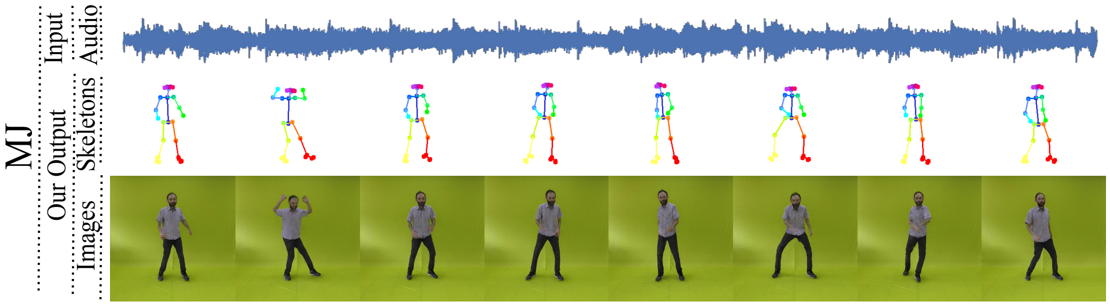


Figure 5.7. Qualitative results using audio sequences for Michael Jackson dance style. First row: input audio; Second row: the sequence of skeletons generated with our method; Third row: the animation of an avatar by vid2vid using our skeletons.

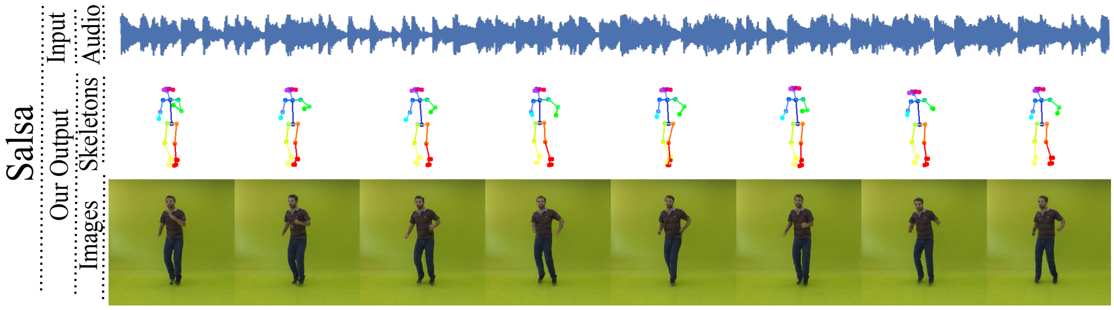


Figure 5.8. Qualitative results using audio sequences for Salsa dance style. First row: input audio; Second row: the sequence of skeletons generated with our method; Third row: the animation of an avatar by vid2vid using our skeletons.

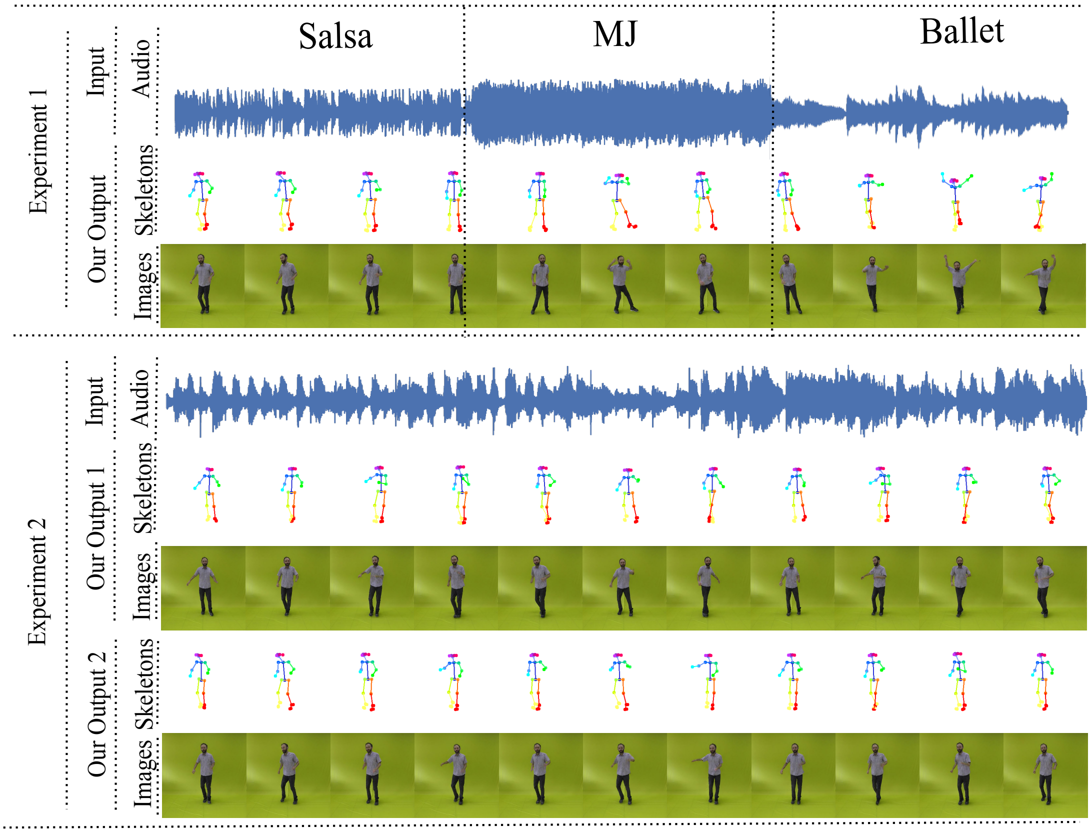


Figure 5.9. Experiment 1 shows the ability of our method to generate different sequences with smooth transition from one given input audio composed of different music styles. Experiment 2 illustrates the responsiveness of our method to the audio style changes.

to the dance style, but they are not present along the whole sequence. For instance, in Figure 5.5, one can see that the last generated skeleton/frame looks like a spin, usually seen in ballet performances, but the previous poses do not indicate any correlation to this dance style. Conversely, our method generates poses commonly associated with ballet movements such as rotating the torso with stretched arms.

Both results for our approach, the sequence of human poses generated by our motion generator, and the video sequence with a virtual avatar are shown in Figures 5.6, 5.7 and 5.8 shows that for all three dance styles, the movement signature was preserved. Another characteristic of our method, shown in Figures 5.6, 5.7 and 5.8, is the ability to synthesize video sequences of different actors.

Moreover, the *Experiment 1* in Figure 5.9 demonstrates that our method is highly responsive to audio style changes since our classifier acts sequentially on subsequent music portions. This enables it to generate videos where the performer executes movements from different styles. In other words, the motion generated varies over time following the classification of the music also over time. Together these results show that the proposed approach holds the ability to create highly discriminative and plausible dance movements.

On the other hand, *Experiment 2* in Figure 5.9 also shows that our method can generate different sequences from one given input audio. Since our model is conditioned on the music style from the audio classification pipeline, and not on the music itself. Therefore it exhibits the capacity of generating varied motions while still preserving the learned motion signature of each dance style. The variability in the movements for the same audio classification is due to variations in the spatio-temporal signal from the Gaussian process as addressed in Section 3.2 and illustrated in Figure 3.5. In other words, for the same input audio, our approach can deliver different motion sequences, and consequently different video sequences, the variability of the motion is related to the gaussian noise from the Gaussian process, and not with the input music itself. Thus, we can generate uncountable samples of motion for the same music, always preserving the motion signature.

Chapter 6

Conclusion

In this thesis, we propose a new method for synthesizing human motion from music. Unlike previous methods, we explore graph convolutional networks trained in an adversarial regime to address the problem. We use the music style information to condition the motion generation and produce realistic human movements with respect to a dance style. We achieved qualitative and quantitative performance as compared to state of the art. Our method outperformed Dancing to Music method proposed by Lee et al. [2019], in terms of FID, GAN-Train, and GAN-Test metrics. We also conducted a user study, which showed that our method received similar scores to real dance movements, which was not observed in the competitor. Moreover, we presented a new dataset with audio and visual data, carefully collected to train and evaluate algorithms designed to synthesize human motion in dance scenarios. We expect our method and the dataset to be one step further towards fostering new approaches for generating human motions.

The work present in this thesis, indicates that, when working with learning techniques or data-driven approaches, the model awareness of the data structure inherent to the motion generation problem can result in simpler models and with better results. Moreover, we found that exploring the relationship between information from the different motion, audio and visual modalities in videos of people dancing, allows us to create an effective approach to produce realistic animations of virtual characters. Finally, we would like to highlight two major points: *i)* The data structure has a key role in learning motion; *ii)* Auditory information can help to solve visual problems, more precisely motion problems, since we explore the relationship between audio data and the motion a human being is performing.

Yet our approach has some limitations: *i)* We condition the motion generation on the audio class, however, if during inference time our audio classifier misclassify the audio, we will condition the motion to the wrong dance style; *ii)* Since we only condition the motion to the style, the variability of the movements in one dance style is related only with the random noise from the Gaussian process. Simultaneously using the audio information to create this variability should be a more suitable approach; *iii)* To create our dataset, we need humans to manually select representatives parts of a video where there is a representative dance performance, which increases the cost to collect data to train to train the proposed model.

6.1 Future Work

The problem of automatic dance generation, and beyond that, the problem of automatic video generation of a virtual avatar performing a dance motion, has several aspects that can be explored to improve the method proposed in this thesis. For instance, we could generate motions in 3D beyond the image space (2D coordinates). One situation where the generation of 3D motions can be attractive is when we are designing a game, where realistic dance motions for the characters should make the game more immersive. For that, there are two major modifications to be done in our approach: *i)* We need to change the dataset annotation to corresponding 3D joints of an avatar compatible with the ones in the game. Automatic data generation should help in this task; *ii)* The network architecture must consider 3D coordinates instead of 2D coordinates. Also modifications in the loss function should be applied since the 3D world space is more complex than the image space.

The most natural improvement in our approach is the extension of the dataset in terms of dance styles. This extension should allow us to stress our approach, and create samples to be used in realistic animations following the auditory data of videos. This can be done by following the pipeline described in Chapter 4.

Another future work is the option to change the video synthesis, which can explore methods designed specifically for the task of video transfer in human motion contexts. This modification can produce better results if the selected method has elements to deal with video transfer of the human motion. As examples of methods that could be used, we can cite the methods proposed by Chan et al. [2019] and Gomes et al. [2020], where the method of Chan et al. [2019] uses a facial GAN to improve results in the face of the virtual avatar in contrast with the method of Wang et al. [2018]. On the other hand, the work of Gomes et al. [2020] allows adding motion restrictions to the movement, making possible realistic interactions with the environment.

Finally, some changes in the designing of our latent space could solve limitations of our method, such as the misclassification of the audio classifier leading to wrong style motion generation, and the issue with temporal constraints of the problem. They could be addressed using the auditory information, since this data has temporal information. By changing our latent space, we could produce results where the final generated motion is more connected to the audio data, in other words, the motion follows better the beat of the music. This could be done by exploring other techniques to deal with temporal constraints as done by Li et al. [2020], who have employed transformers to solve the temporal constraints of the problem.

Bibliography

Andriluka, M., Pishchulin, L., Gehler, P., and Schiele, B. (2014). 2D human pose estimation: New benchmark and state of the art analysis. In *CVPR*, pages 3686–3693.

Arandjelovic, R. and Zisserman, A. (2018). Objects that sound. In *European Conference on Computer Vision (ECCV)*, pages 435--451.

Aytar, Y., Vondrick, C., and Torralba, A. (2016). Soundnet: Learning sound representations from unlabeled video. In *Advances in neural information processing systems*, pages 892--900.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41--48.

Bregler, C., Covell, M., and Slaney, M. (1997). Video rewrite: Driving visual speech with audio. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, page 353–360, USA. ACM Press/Addison-Wesley Publishing Co.

Cao, Z., Hidalgo Martinez, G., Simon, T., Wei, S., and Sheikh, Y. A. (2019). Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1.

Cao, Z., Simon, T., Wei, S., and Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1302–1310.

Chan, C., Ginosar, S., Zhou, T., and Efros, A. (2019). Everybody dance now. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5932–5941.

Chen, Y., Shen, C., Wei, X., Liu, L., and Yang, J. (2017). Adversarial posenet: A structure-aware convolutional network for human pose estimation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1221–1230.

Choi, K.-J. and Ko, H.-S. (2000). On-line motion retargeting. *Journal of Visualization and Computer Animation*, 11:223–235.

Cudeiro, D., Bolkart, T., Laidlaw, C., Ranjan, A., and Black, M. J. (2019). Capture, learning, and synthesis of 3d speaking styles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10101--10111.

De Boor, C., De Boor, C., Mathématicien, E.-U., De Boor, C., and De Boor, C. (1978). *A practical guide to splines*, volume 27. springer-verlag New York.

Ebel, R. and Frisbie, D. (1991). *Essentials of Educational Measurement.* Prentice Hall.

Fang, H., Xie, S., Tai, Y., and Lu, C. (2017). Rmpe: Regional multi-person pose estimation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2353–2362.

Ferreira, J. P., Coutinho, T. M., Gomes, T. L., Neto, J. F., Azevedo, R., Martins, R., and Nascimento, E. R. (2020). Learning to dance: A graph convolutional adversarial network to generate realistic dance motions from audio. *Computers & Graphics.*

Fragkiadaki, K., Levine, S., Felsen, P., and Malik, J. (2015). Recurrent network models for human dynamics. In *ICCV*, pages 4346–4354.

Ghosh, P., Song, J., Aksan, E., and Hilliges, O. (2017). Learning human motion models for long-term predictions. In *2017 International Conference on 3D Vision (3DV)*, pages 458–466.

Ginosar, S., Bar, A., Kohavi, G., Chan, C., Owens, A., and Malik, J. (2019). Learning individual styles of conversational gesture. In *CVPR.*

Gleicher, M. (1998). Retargetting motion to new characters. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pages 33--42, New York, NY, USA. ACM.

Gomes, T. L., Martins, R., Ferreira, J., and Nascimento, E. R. (2020). Do as i do: Transferring human motion and appearance between monocular videos with spatial and temporal constraints. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 3355–3364.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672--2680.

Gui, L.-Y., Wang, Y.-X., Liang, X., and Moura, J. M. (2018). Adversarial geometry-aware human motion prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 786--803.

Güler, R. A., Neverova, N., and Kokkinos, I. (2018). Densepose: Dense human pose estimation in the wild. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7297–7306.

Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., and Ng, A. Y. (2014). Deep speech: Scaling up end-to-end speech recognition.

Hershey, S., Chaudhuri, S., Ellis, D. P. W., Gemmeke, J. F., Jansen, A., Moore, C., Plakal, M., Platt, D., Saurous, R. A., Seybold, B., Slaney, M., Weiss, R., and Wilson, K. (2017). CNN architectures for large-scale audio classification. In *ICASSP*.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pages 6626--6637.

Huang, R., Hu, H., Wu, W., Sawada, K., and Zhang, M. (2020). Dance revolution: Long sequence dance generation with music via curriculum learning. *arXiv preprint arXiv:2006.06119*.

Ikeuchi, K., Ma, Z., Yan, Z., Kudoh, S., and Nakamura, M. (2018). Describing upper-body motions based on labanotation for learning-from-observation robots. *International Journal of Computer Vision*, 126(12):1415--1429.

Jain, A., Zamir, A. R., Savarese, S., and Saxena, A. (2016). Structural-rnn: Deep learning on spatio-temporal graphs. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5308–5317.

Jang, D.-K. and Lee, S.-H. (2020). Constructing human motion manifold with sequential networks. *Computer Graphics Forum*.

Kanazawa, A., Zhang, J. Y., Felsen, P., and Malik, J. (2019). Learning 3d human dynamics from video. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5607–5616.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2018). Progressive growing of GANs for improved quality, stability, and variation. In *ICLR*.

Kim, H. J. and Lee, S.-H. (2019). Perceptual characteristics by motion style category. In *Eurographics (Short Papers)*, pages 1--4.

Kipf, T. N. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *ICLR*.

Kocabas, M., Athanasiou, N., and Black, M. J. (2020). Vibe: Video inference for human body pose and shape estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kolotouros, N., Pavlakos, G., Black, M., and Daniilidis, K. (2019). Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2252–2261.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22(1):79--86.

Lee, H.-Y., Yang, X., Liu, M.-Y., Wang, T.-C., Lu, Y.-D., Yang, M.-H., and Kautz, J. (2019). Dancing to music. In *Advances in Neural Information Processing Systems*, pages 3586--3596.

Leman, M. (2014). The role of embodiment in the perception of music. *Empirical Musicology Review*, 9(3-4):236--246.

Li, J., Wang, C., Zhu, H., Mao, Y., Fang, H., and Lu, C. (2019). Crowdpose: Efficient crowded scenes pose estimation and a new benchmark. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10855–10864.

Li, J., Yin, Y., Chu, H., Zhou, Y., Wang, T., Fidler, S., and Li, H. (2020). Learning to generate diverse dance motions with transformer. *arXiv preprint arXiv:2008.08171*.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740--755. Springer.

Liu, L. and Hodgins, J. (2018). Learning basketball dribbling skills using trajectory optimization and deep reinforcement learning. *ACM Trans. Graph.*, 37(4).

Luger, S. K. K. and Bowles, J. (2016). Comparative methods and analysis for creating high-quality question sets from crowdsourced data. In Markov, Z. and Russell, I., editors, *Proceedings of the Twenty-Ninth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2016, Key Largo, Florida, USA, May 16-18, 2016*, pages 185--190. AAAI Press.

Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579--2605.

Martinez, J., Black, M. J., and Romero, J. (2017). On human motion prediction using recurrent neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4674–4683.

Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, page III–1310–III–1318. JMLR.org.

Peng, X. B., Berseth, G., Yin, K., and van de Panne, M. (2017). Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning. *ACM Transactions on Graphics (Proc. SIGGRAPH 2017)*, 36(4).

Peng, X. B., Kanazawa, A., Malik, J., Abbeel, P., and Levine, S. (2018). SFV: Reinforcement learning of physical skills from videos. *ACM Trans. Graph.*, 37(6).

Rasmussen, C. E. (2003). Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63--71. Springer.

Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., and Lee, H. (2016). Generative adversarial text to image synthesis. In Balcan, M. F. and Weinberger, K. Q., editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1060--1069, New York, New York, USA. PMLR.

Ren, X., Li, H., Huang, Z., and Chen, Q. (2019). Music-oriented dance video synthesis with pose perceptual loss. *arXiv preprint arXiv:1912.06606*.

Shiratori, T. and Ikeuchi, K. (2008). Synthesis of dance performance based on analyses of human motion and music. *Information and Media Technologies*, 3(4):834--847.

Shlizerman, E., Dery, L., Schoen, H., and Kemelmacher-Shlizerman, I. (2018). Audio to body dynamics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7574--7583.

Shmelkov, K., Schmid, C., and Alahari, K. (2018). How good is my GAN? In *ECCV*, pages 213--229.

Simon, T., Joo, H., Matthews, I., and Sheikh, Y. (2017). Hand keypoint detection in single images using multiview bootstrapping. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4645–4653.

Smith, H. J., Cao, C., Neff, M., and Wang, Y. (2019). Efficient neural networks for real-time motion style transfer. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 2(2):1--17.

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. In *9th ISCA Speech Synthesis Workshop*, pages 125--125.

Villegas, R., Yang, J., Ceylan, D., and Lee, H. (2018). Neural kinematic networks for unsupervised motion retargetting. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Villegas, R., Yang, J., Zou, Y., Sohn, S., Lin, X., and Lee, H. (2017). Learning to generate long-term future via hierarchical prediction. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3560--3569. JMLR. org.

Wang, H., Ho, E. S. L., Shum, H. P. H., and Zhu, Z. (2019). Spatio-temporal manifold learning for human motions via long-horizon modeling. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–1.

Wang, Q., Artières, T., Chen, M., and Denoyer, L. (2020). Adversarial learning for modeling human motion. *The Visual Computer*, 36(1):141--160.

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Liu, G., Tao, A., Kautz, J., and Catanzaro, B. (2018). Video-to-video synthesis. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 1152--1164.

Wang, X., Chen, Q., and Wang, W. (2014). 3d human motion editing and synthesis: a survey. *Computational and Mathematical Methods in Medicine*, 2014:104535--104535.

Wei, S., Ramakrishna, V., Kanade, T., and Sheikh, Y. (2016). Convolutional pose machines. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4732.

Weiss, C. (2005). FSM and k-nearest-neighbor for corpus based video-realistic audio-visual synthesis. In *INTERSPEECH 2005 - Eurospeech, 9th European Conference on Speech Communication and Technology, Lisbon, Portugal, September 4-8, 2005*, pages 2537--2540. ISCA.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. (2020). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*.

Xia, S., Wang, C., Chai, J., and Hodgins, J. (2015). Realtime style transfer for unlabeled heterogeneous human motion. *ACM Transactions on Graphics (TOG)*, 34(4):1--10.

Xiu, Y., Li, J., Wang, H., Fang, Y., and Lu, C. (2018). Pose Flow: Efficient online pose tracking. In *British Machine Vision Conference (BMVC)*.

Yan, S., Li, Z., Xiong, Y., Yan, H., and Lin, D. (2019). Convolutional sequence generation for skeleton-based action synthesis. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4393–4401.

Yan, S., Xiong, Y., and Lin, D. (2018). Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI Conference on Artificial Intelligence (AAAI)*.

Ye, Z., Wu, H., Jia, J., Bu, Y., Chen, W., Meng, F., and Wang, Y. (2020). Choreonet: Towards music to dance synthesis with choreographic action unit. *arXiv preprint arXiv:2009.07637*.

Zhuang, W., Wang, Y., Robinson, J., Wang, C., Shao, M., Fu, Y., and Xia, S. (2020). Towards 3d dance motion synthesis and control. *arXiv preprint arXiv:2006.05743*.