Application of Generative Adversarial Networks for Generation and Classification of Human Movements

A Thesis

Presented in Partial Fulfillment of the Requirements for the

Degree of Master of Science

with a

Major in Computer Science

in the

College of Graduate Studies

University of Idaho

by

Longze Li

Major Professors: Aleksandar Vakanski, Ph.D.; Min Xian, Ph.D.

Committee Members: Robert Hiromoto, Ph.D.; Xiaogang Ma, Ph.D.

Department Administrator: Terry Soule, Ph.D.

May 2020

Authorization to Submit Thesis

This thesis of Longze Li, submitted for the degree of Master of Science with a Major in Computer Science and titled "Application of Generative Adversarial Networks for Generation and Classification of Human Movements," has been reviewed in final form. Permission, as indicated by the signatures and dates below, is now granted to submit final copies to the College of Graduate Studies for approval.

Major Professors: ______________________________ Date: __________

Aleksandar Vakanski, Ph.D.

______________________________ Date: __________

Min Xian, Ph.D.

Committee Members: ______________________________ Date: __________

Robert Hiromoto, Ph.D.

______________________________ Date: __________

Xiaogang Ma, Ph.D.

Department

Administrator: ______________________________ Date: __________

Terry Soule, Ph.D.

Abstract

This thesis proposes a method for mathematical modeling of human movements by using deep artificial neural networks, with application in modeling patient exercise episodes performed during physical therapy and rehabilitation sessions. The generative adversarial network (GAN) structure is adopted, whereby a discriminative and a generative model are trained concurrently in an adversarial manner. The capacity of GAN models for generating synthetic data offers a potential to artificially augment the size of datasets for biomedical applications, where collecting large datasets is notoriously challenging, due to the need for access to patients, as well as due to privacy, safety, and ethics concerns. Synthetically augmented datasets have demonstrated improved robustness and overall performance of machine learning models across various data formats and modalities. The thesis examines different network architectures, with the discriminative and generative models structured as deep subnetworks of hidden layers comprised of convolutional or recurrent computational units. The models are validated on a dataset of physical rehabilitation movements recorded with an optical motion tracker. The results demonstrate an ability of GAN network architectures for generation of movement examples that resemble the recorded rehabilitation movement sequences, and for classification of unseen instances of the movements.

Acknowledgments

First and foremost, I would like to thank the members of my graduate committee. I was fortunate to have an opportunity to work with Dr. Vakanski even before coming to the University of Idaho. He has always been very inspiring, patient, and supportive during my transition from mechanical engineering to computer science degrees, and in my graduate study. I thank Dr. Xian’s expertise and vision that helped me start the research in machine learning. I’ve taken three classes from Dr. Hiromoto from different areas in computer science, and all of the classes broadened my knowledge in computer science. I also had a chance to work with him on a cryptography project, and it deepened my knowledge in cryptography and encryption. All of three members above are great mentors to me. I took Dr. Ma’s data science class; this class is extremely useful, and it helped me obtaining a data science internship last summer.

Dedication

Most importantly, I would like to thank my parents and dedicate this thesis to them. Without their support, both mentally and financially, I would have never had the chance to pursue my Master of Science degree at the University of Idaho.

Table of Contents

List of Tables

List of Figures

CHAPTER 1: INTRODUCTION

1.1 Physical Rehabilitation and Patient Compliance

Patients recovering from stroke, surgery, nerve damage or bone fracture are regularly enrolled in physical therapy and rehabilitation programs to regain muscle strength, relieve pain, and improve range of motion. Both long–term and short–term physical therapy provide positive results in treating musculoskeletal trauma and functional movement disorders [1], [2]. However, rehabilitation treatment imposes a substantial economic burden on patients and healthcare systems [3]–[5]. For instance, the annual cost of physical rehabilitation programs in the US exceeds 13.5 billion dollars, based on the Medical Expenditure Panel Survey generated by the US federal government [4]. The annual expenditure was produced by nearly 9 million adults during approximately 88 million physical rehabilitation episodes.

In physical therapy and rehabilitation programs, clinicians prescribe to patients a set of exercises and task them with performing a recommended number of repetitions of the exercises for a certain period of time. The patients typically perform the prescribed regimen initially in a clinical setting under direct supervision by a medical professional. This type of rehabilitation treatment is restricted by the availability of trained clinicians and it places demands on patients' schedules. To increase the flexibility of rehabilitation programs, home-based rehabilitation is often employed as a subsequent supplement to clinic-based programs. In home-based regimens, patients perform the recommended exercises in their residence following the given instructions by the clinician, while recording their daily progress in a logbook; the patients visit the clinic periodically for progress assessment.

The efficiency of physical rehabilitation programs is highly related to patient adherence to prescribed exercises [6]. On the other hand, in a home-based setting, it is difficult to determine if a patient complies with the therapy program, because most of the patients do not acknowledge the incompliance [7]. Indeed, medical sources report low levels of patient motivation and adherence to the prescribed exercise regimens in home-based rehabilitation, which results in prolonged treatment duration and increased healthcare costs [6], [8]. Although many factors that reduce patient motivation and engagement in rehabilitation training have been identified, the lack of timely feedback and real-time supervision by a healthcare professional in an at-home setting is often cited as the most influential factor [9]. Poor motivation and supervision promote further risk because patients may perform exercises incorrectly as a result of those factors, which increases the risk of re-injury [6].

1.2 Movement Modeling using Machine Learning

The latest progress in machine learning offers a potential to identify incorrect performance of physical rehabilitation exercises and provide instantaneous feedback to the patient. Further, it can also provide a basis for the healthcare professionals to be proactive and take early corrective actions, if needed. Efficient application of machine learning for evaluation of patient performance requires corresponding datasets of therapy movements for algorithm training purposes, and formulation of robust mathematical models of human body trajectories executed during physical therapy exercises.

Modeling human movements has been an essential research topic in various fields and disciplines. Congruent models of human movements furnish great benefits to ergonomic design [10], visual surveillance [11], transfer of human skills to robotic learning systems [12], etc. However, mathematical modeling of human movements remains an open research problem, due to the challenges associated with the complex stochastic and nonlinear character of the data. First, human movements are inherently random, because of the stochastic nature of processing motory commands by the brain (e.g., we cannot re-create identical movements or draw perfectly straight lines). Second, human movements are marked with a high degree of variability across individuals' characteristics, such as age, gender, weight, fatigue, or even pain. Third, the uncertainties introduced by sensory measurement and processing errors add to the complexity in movement modeling.

A current trend in machine learning related to the implementation of *deep artificial neural networks* (NNs) for modeling and representation of complex nonlinear data across various domains [13] has paved a promising path to human motion modeling. Within the published literature on modeling human movements using machine learning approaches, most works focus on recognition and classification of movements into a particular movement type. To that end, a variety of traditional machine learning algorithms have been applied, including support vector machines, hidden Markov models, and *k*-nearest neighbors. In recent years, a body of research emerged based on the implementation of artificial NNs for the task at hand. Encoder-decoder NNs have been a commonly employed means for extraction of salient attributes in movement trajectories of captured skeletal data [14], [15]. NNs with convolutional computational units have been designed for recognition of human movements, for example, in surveillance videos [16]. Another network architecture that employs recurrent connections between the computational units has been extensively used for modeling sequential data in general [17], [18], and human motions in particular [19], [20]. Beside for movement classification task, machine learning methods have also been employed for prediction of future motion patterns, e.g., fall detection in seniors [21], or automated anticipation of driver activities [22].

Analogously, in the domain of physical therapy and rehabilitation several researchers employed machine learning for classification of patient movements [23] and for counting the number of repetitions in each exercise [24]. In reference [25], an intelligent robotic assistant employs machine learning for planning the next therapy session based on the patient’s current progress. Similarly, machine learning-based assistants have been integrated into virtual reality therapy systems for monitoring patient performance and customizing the treatment plan according to the patient’s progress [26]–[28]. In the treatment of phantom limb pain, it was found that the combination of machine learning, augmented reality, and gaming produces improved outcomes in comparison to traditional treatment approaches [29]. Another class of therapy tools employs a motion capturing camera and it displays in real-time on a screen the executed movements by the patient, and simultaneously a graphical avatar is displayed on the side of the screen that demonstrates the correctly performed movements as recommended by the physical therapist [30], [31]. These tools are excellent examples of innovative solutions and systems in support of home-based physical therapy, as they can potentially improve patient adherence to prescribed therapy programs, and subsequently, lead to reduced rehabilitation period, reduced time to functional recovery, and reduced healthcare costs.

1.3 Motion Capture Systems

To address the challenges associated with home-based and in-clinic rehabilitation programs, the development of systems that can reliably capture human movements is crucial. Although standard vision cameras have been used as a motion sensor in several related works, they provide only 2-dimensional information about the captured scene and the lack of the third dimension’s information imposes limits on the evaluation accuracy. To cope with this deficiency, optical motion tracking systems have often been used for this task. These systems employ a set of markers attached to strategic locations on a patient’s body that are tracked by multiple high-resolution cameras. Optical motion trackers rely on computational algorithms to reconstruct the 3-dimensional scene by comparing and aligning the images taken by the set of multiple cameras. Furthermore, recent technology for 3-dimensional scene reconstruction based on vison/depth cameras has become popular due to the low cost and ease of use. Among the commercial vision/depth sensors, Microsoft Kinect has been the preferred choice in most related works. Inertial sensors and accelerometers have also been extensively used for motion tracking and evaluation, due to their low cost and simple principles of operation. The provision of low-cost sensors with integrated functionality for tracking human motions furnishes an opportunity for the development of home-based systems for rehabilitation programs. Combined with efficient computational algorithms for modeling and analyzing human motions, these technologies can play an important role in the future management of rehabilitation regimens and for monitoring patient performance and progress.

1.4 GANs for Movement Modeling

This thesis presents a novel method for modeling and evaluation of physical rehabilitation exercises based on an NN architecture known as *Generative Adversarial Networks* (GANs). Introduced by Goodfellow *et al*. in 2014 [32], GANs represent a deep learning model comprised of two competitive subnetworks: a generative subnetwork (commonly referred to as a generator) and a discriminative subnetwork (i.e., a discriminator). The two subnetworks are trained in an adversarial mode, where the generator improves in producing data that resemble the real input data, and the discriminator improves in distinguishing real input data from the data samples provided by the generator. GAN models have had tremendous success in the domain of image processing, e.g., for generating super resolution photo-realistic images from text [33], face aging images in entertainment [34], blending of objects from one picture into the background of another picture, as well as in other applications, such as generating hand-written text, and music sequence generation [35].

This thesis investigates the capacity of GAN models for generating human movement data related to physical therapy exercises. It was motivated by the research by Hyland *et al*. [36]

where the authors designed a GAN model for generating synthetic medical data resembling the records from an intensive care unit. In general, almost all research on GANs is directed toward generating images, and only a few works have applied GANs for generating time-series data. On the other hand, the provision of means for synthesizing realistic time-series data can benefit several application areas. For the considered problem, the ability to produce movement sequences that resemble patient therapy exercises has a potential to augment the datasets of recorded therapy exercises and to lead to improved movement models. Consequently, this work presents an evaluation of different GAN architectures for generating synthetic movement sequences. Additionally, the performance of GAN networks for assessment of the level of correctness of therapy movements is evaluated. For that purpose, soft labels are introduced for the movement repetitions based on the average deviation from a set of consistently performed movements. The study found that GANs are suitable for both generation and evaluation of therapy movement sequences.

1.5 Thesis Organization

The thesis is organized as follows. Chapter 2 introduces GAN models and provides an overview of several GAN architectures that are relevant for the considered task of rehabilitation episodes. Chapter 3 describes the movement data related to physical therapy exercises that are used for training and validation of the NNs. The investigated architectures of the GAN models are presented in Chapter 4. Chapter 4 also presents the experimental results of using GANs for generating movement data and for evaluating exercise performance. Chapter 5 briefly summarizes the work and concludes the thesis.

CHAPTER 2: GENERATIVE ADVERSARIAL NETWORKS

2.1 GANs Basics

As stated in the Introduction chapter, GANs consist of two subnetworks: a *discriminator D*, and a *generator G* subnetwork. The discriminator maps the input data to class probabilities, i.e., it models the probability distribution of the output labels conditioned on the input data. On the other hand, the generator models the probability distribution of the input data, which allows generating new data instances by sampling from the model distribution. Both subnetworks D and G are trained simultaneously in an adversarial manner, where the generator G attempts to improve in creating synthetic data that approximate the input data, and the discriminator D attempts to improve in differentiating the real data from the synthetically generated data.

Let x denotes the input to the network, where $x \sim \mathbb{P}_r$, and $\mathbb{P}_r$ denotes the probability distribution of the real input data. The goal of the generator in GANs is to learn a model distribution $\mathbb{P}_g$ that approximates the unknown distribution of the real data $\mathbb{P}_r$. For that purpose, a random variable z sampled from a fixed (e.g., uniform or Gaussian) probability distribution is used as the input to the generator, as illustrated in Figure 2.1. During the training phase, the parameters of the generator are iteratively varied in order to reduce the distance, or divergence, between the distributions $\mathbb{P}_g$ and $\mathbb{P}_r$. The output of the generator is denoted $\bar{x}$ here, i.e., the generator mapping is $G : z \mapsto \bar{x}$.

To solve the described problem, a network loss function H is introduced in the form of a cross-entropy,

$$H(D,G) = \underset{x \sim \mathbb{P}_r}{\mathbb{E}}\left[\log\left(D(x)\right)\right] + \underset{\bar{x} \sim \mathbb{P}_g}{\mathbb{E}}\left[\log\left(1 - D(\bar{x})\right)\right]. \quad (1)$$

In the above equation $\mathbb{E}_{\mathbb{P}}[\cdot]$ is the expected value operator with respect to a distribution $\mathbb{P}$, $D(x)$ is the output of the discriminator subnetwork on real input data, and $D(\bar{x})$ denotes the output of the discriminator on synthetically generated data.

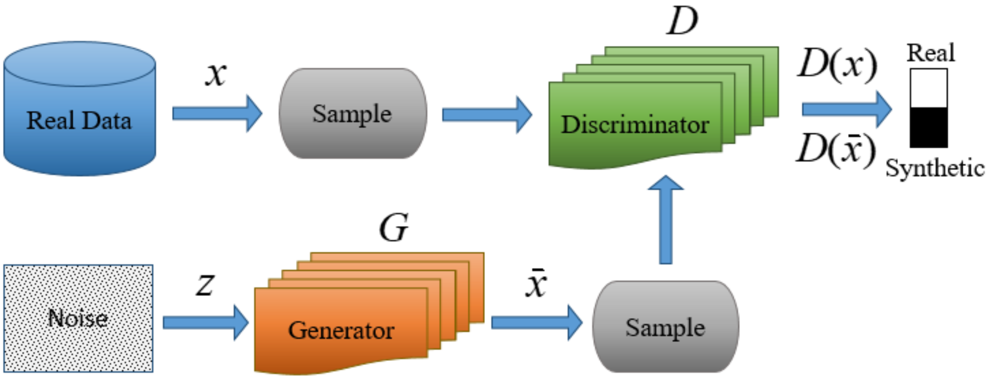


Figure 2.1. A GAN model consists of a generator and a discriminator. The generator takes random noise as input and attempts to produce synthetic data $\bar{x}$ that resemble the real data x . The discriminator attempts to discriminate real data from the synthetic data produced by the generator.

The discriminator is trained to maximize the loss function *H*, and the generator is trained to minimize the loss function *H*, i.e., the goal is

$$\min_{G} \max_{D} H(D,G). \tag{2}$$

In the game theory, this is called a minimax game. The two subnetworks are trained in a competitive two-player scenario, where both the generator and discriminator improve their performance until a Nash equilibrium is reached. One can note that minimizing the function in Eq. (1) is equivalent to minimizing the Jensen-Shannon (JS) divergence between the real data distribution $\mathbb{P}_r$ and the model distribution $\mathbb{P}_g$.

In the case of binary classification, the discriminator is trained to maximize *H* by forcing $D(x)$ to approach 1 and $D(\bar{x})$ to approach 0 (Figure 2.1). Contrarily, the generator is trained to minimize *H* by forcing $D(\bar{x})$ to approach 1. Backpropagation is employed for updating the parameters of both the discriminator and generator during training, with the distribution $\mathbb{P}_g$ becoming more and more similar to $\mathbb{P}_r$.

The main disadvantage of GANs is the training instability. More specifically, if the generator is trained faster than the discriminator, a mode collapse (also known as a Helvetica scenario) can occur, where the generator maps many values of the random variable *z* to the same value of *x*, and reduces its capacity to learn the distribution of the real data $\mathbb{P}_r$. Besides, the model does not allow for explicit calculation of $G(x)$, and as a result, the quality of the generated data (e.g., images, as the most common data in GANs) is typically evaluated by visual observation and comparison to the actual input data. Another shortcoming of GANs is the presence of noise (and blur in the case of image data), due to the introduced random noise *z* as input to the generator.

Based on the instant success of the GANs model, a large number of GAN variants have been developed since the original work. A body of works addressed some of the above-described shortcomings [37]–[40], and other variants were designed specifically for domain-specific solutions [41], [42]. For example, in conditional GANs [41] the data is conditioned on the class labels, which allows generating images with the desired class (e.g., specific digits of the MNIST dataset). InfoGAN introduced a novel type of regularization in GANs based on the mutual information learned between the incompressible noise component and the latent code of the generator. BiGAN [43] introduced an encoder sub-network to the original generator-discriminator architecture, which allows projecting the data back to the latent space. The authors demonstrated that the ability to obtain the inverse mapping was useful for learning improved feature representations. BigGAN [44] applied orthogonal regularization to the generator which improved the scalability of the model, as well as the authors demonstrated that truncating the latent space resulted in improved robustness.

Consequently, the proposed architecture reduced the training instabilities while showing a capacity for generating impressive synthetic images. CycleGAN [45] was designed for transforming images from one domain to another. The transformation is achieved by two GANs that are trained in such a way that consistency is maintained in the transformation of an image from one domain to another domain, as well as in the backward transformation of images from the target domain to the source domain. Such transformation is referred to as being cycle consistent.

In the ensuing subsections, a brief overview of several GAN architectures is presented that are relevant for the considered problem of modeling time-series data related to patient therapy movement episodes.

2.2 Deep Convolutional GANs

Deep Convolutional GANs (DCGANs) [37] introduce several constraints and modifications to the original GAN architecture for improved stability and performance. As the name implies, the generator and discriminator subnetworks are composed of multiple layers of convolutional computational units, as opposed to the multilayer perceptron (MLP) networks proposed in the original GAN paper [32]. The modifications in DCGANs are as follows. First, the network structure in DCGANs replaces pooling layers with strided convolutions, which allows the subnetworks to adjust the spatial down-sampling and up-sampling based on the input data. Second, it eliminates fully connected layers that are commonly used after convolutional layers in deep NNs, and it relies solely on convolutional layers. Third, the DCGANs model employs batch normalization, to stabilize the gradients increase during training and reduce the possibility of a mode collapse. Batch normalization is applied to all layers, except for the output layer of the generator and the input layer of the discriminator. Fourth, ReLU activation function is used for all layers in the generator, except for the last layer where a Tanh activation function is applied. For the discriminator, leaky ReLU activation function is suggested for all layers. By applying the above recommendations, the authors have demonstrated improved classification performance on various datasets of images, and capabilities of generating complex and visually realistic images.

2.3 Wasserstein GANs

Wasserstein GANs (WGANs) [38] introduce a new loss function for training the generator and discriminator subnetworks. The loss function is based on the Wasserstein distance (also known as Earth Mover distance) between the real data distribution $\mathbb{P}_r$ and the model distribution $\mathbb{P}_g$ learned by the generator,

$$W\left(\mathbb{P}_r,\mathbb{P}_g\right)=\inf_{\gamma\in\Pi\left(\mathbb{P}_r,\mathbb{P}_g\right)}\mathop{\mathbb{E}}_{(x,y)\sim\gamma}\left[\|x-y\|\right]. \tag{3}$$

In Eq. (3), $\Pi\left(\mathbb{P}_r,\mathbb{P}_g\right)$ denotes the set of joint distributions $\gamma\left(x,y\right)$ whose marginals are $\mathbb{P}_r$ and $\mathbb{P}_g$. In simpler terms, $\gamma\left(x,y\right)$ defines the amount of earth mass that needs to be moved from a point *x* to a point *y* in order $\mathbb{P}_r$ and $\mathbb{P}_g$ to be identical. Accordingly, the proposed loss function is derived as an approximation to the Wasserstein distance

$$H\left(D,G\right)=\mathop{\mathbb{E}}_{x\sim\mathbb{P}_r}\left[D\left(x\right)\right]-\mathop{\mathbb{E}}_{\bar{x}\sim\mathbb{P}_g}\left[D\left(\bar{x}\right)\right]. \tag{4}$$

Such distance function induces a weaker topology than the Jensen-Shannon (JS) divergence used in the original GANs and given in Eq. (1), and the Kullback-Leibler (KL) divergence commonly used in maximum likelihood estimation. The weaker topology provides a lever for the convergence of the probability distribution of the model $\mathbb{P}_g$ to the real distribution of the data $\mathbb{P}_r$. If the discriminator $D\left(x\right)$ is a *K*- Lipschitz function, it was proven that the proposed loss function in Eq. (4) is continuous and differentiable, and produces stable gradients during training, thereby improving the problem of training instability in GANs.

In addition, the values of the adopted loss function H in Eq. (4) are correlated to the quality of the generated data samples by the generator, and with that WGANs provide a basis for quantifying the performance of the generator, rather than relying on visual observation of the generated samples. Accordingly, during the network training, the loss function in Eq. (4) is used to evaluate the training convergence, i.e., to identify if the network is being trained.

To enforce a Lipschitz constraint on the discriminator, it was proposed to apply clipping of the network parameters into a range $\left[-c, +c\right]$ after each gradient update, where c is a referred to as a clipping constant. The suggested value for c in the WGANs paper [38] is 0.01.

Unlike GANs, the output of the discriminator in WGANs is not a probability; instead, it is an estimate of the Wasserstein distance between the distributions. Therefore, the authors use the term critic in the article, rather than discriminator, due to the similarity with the actor-critic methods in reinforcement learning.

2.4 Recurrent GANs

Recurrent GANs (RGANs) [36] are an alternative GAN model that is designed for handling multi-dimensional time-series data. For that purpose, recurrent computational units are employed for the discriminator and generator. More specifically, a layer of unidirectional Long Short-Term Memory (LSTM) computational units [17] is used for both subnetworks in the RGANs paper.

The proposed approach with RGANs was applied to medical records data from an intensive care unit. The authors investigated the ability of RGANs to generate synthetic medical data samples and the potential for use in data augmentation in cases of insufficiency of real data for training deep learning models. In the article, RGANs were also implemented for processing synthetic sine waves sequences, as well as images. The authors claimed that RGANs are more suitable for dealing with time-series data in comparison to the proposed GANs alternatives composed of layers of convolutional kernels.

CHAPTER 3: PHYSICAL REHABILITATION MOVEMENTS DATA SET

3.1 Data Description

The introduced GAN models are validated on the University of Idaho – Physical Rehabilitation Movements Data (UI–PRMD) set. The full description of the dataset is provided in [46]. A group of 10 healthy subjects performed 10 repetitions for 10 rehabilitation movements. In addition, the subjects performed 10 repetitions for each movement in an incorrect fashion, simulating performance by patients enrolled in physical therapy programs. The data collection for UI–PRMD was approved by the Institutional Review Boards at the University of Idaho on April 26, 2017, under the identification code IRB 16-124. The movement data were collected in the Integrated Sports Medicine Movement Analysis Laboratory (ISMMAL) with the Department of Movement Sciences at the University of Idaho. The movement data were categorized, organized and posted on a dedicated web site for free public access. Potential benefits of publicly posting the UI-PRMD set include the potential to serve as a benchmark for comparison of future research in physical therapy and rehabilitation, and to streamline the process of establishing consistent metrics for evaluation of patient progress in rehabilitation programs.

The demographic information of the 10 subjects who participated in the data collection is provided in Table 3.1. The average age of the subjects was 29.3 years, with a standard deviation of 5.85 years. The exclusion criteria included musculoskeletal injuries, pregnancy, neurological disorders that affect balance, less than 6 months post-orthopedic surgery, less than 2 months post-visceral surgery, contagious illnesses, and taking medications that affect proprioceptive capabilities. In addition, the study did not include children under the age of 18.

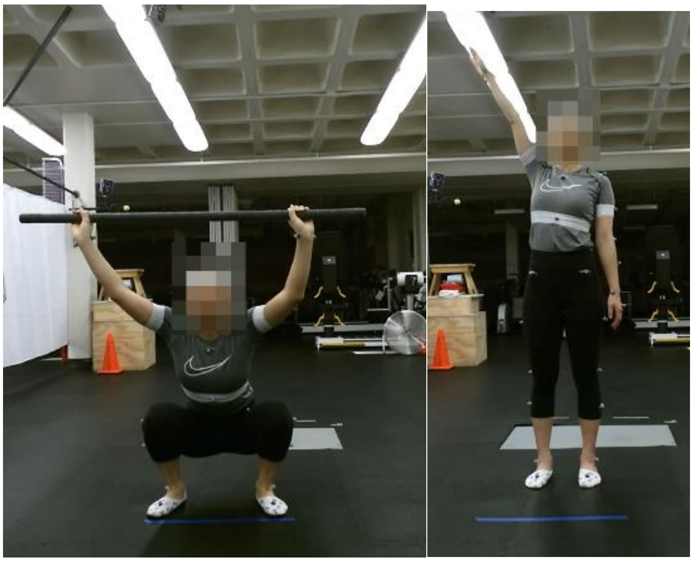
Table 3.1. Demographic information for the subjects.

Subject ID	Gender	Height (cm)	Weight (kg)	BMI	Dominant side	Age
s01	Female	169.0	69.4	24.3	Right	23
s02	Male	180.0	83.0	25.6	Right	31
s03	Male	169.5	64.8	22.6	Right	44
s04	Female	178.5	79.4	24.9	Right	31
s05	Male	185.5	148.6	43.2	Right	28
s06	Female	164.6	53.6	19.8	Right	27
s07	Female	166.1	53.1	19.2	Left	24
s08	Male	170.5	77.3	26.6	Right	29
s09	Female	164.0	56.0	20.8	Right	26
s10	Male	174.2	94.7	31.2	Left	26

The recorded time-series sequences related to two common training movements in physical therapy exercises—a deep squat, hereafter Movement 1, and a standing shoulder abduction, hereafter Movement 2—are used in this work. A brief description of the two movements is provided in Table 3.2. Examples of the performed movements by one of the subjects are shown in Figure 3.1.

Table 3.2. Movement description and incorrect performance.

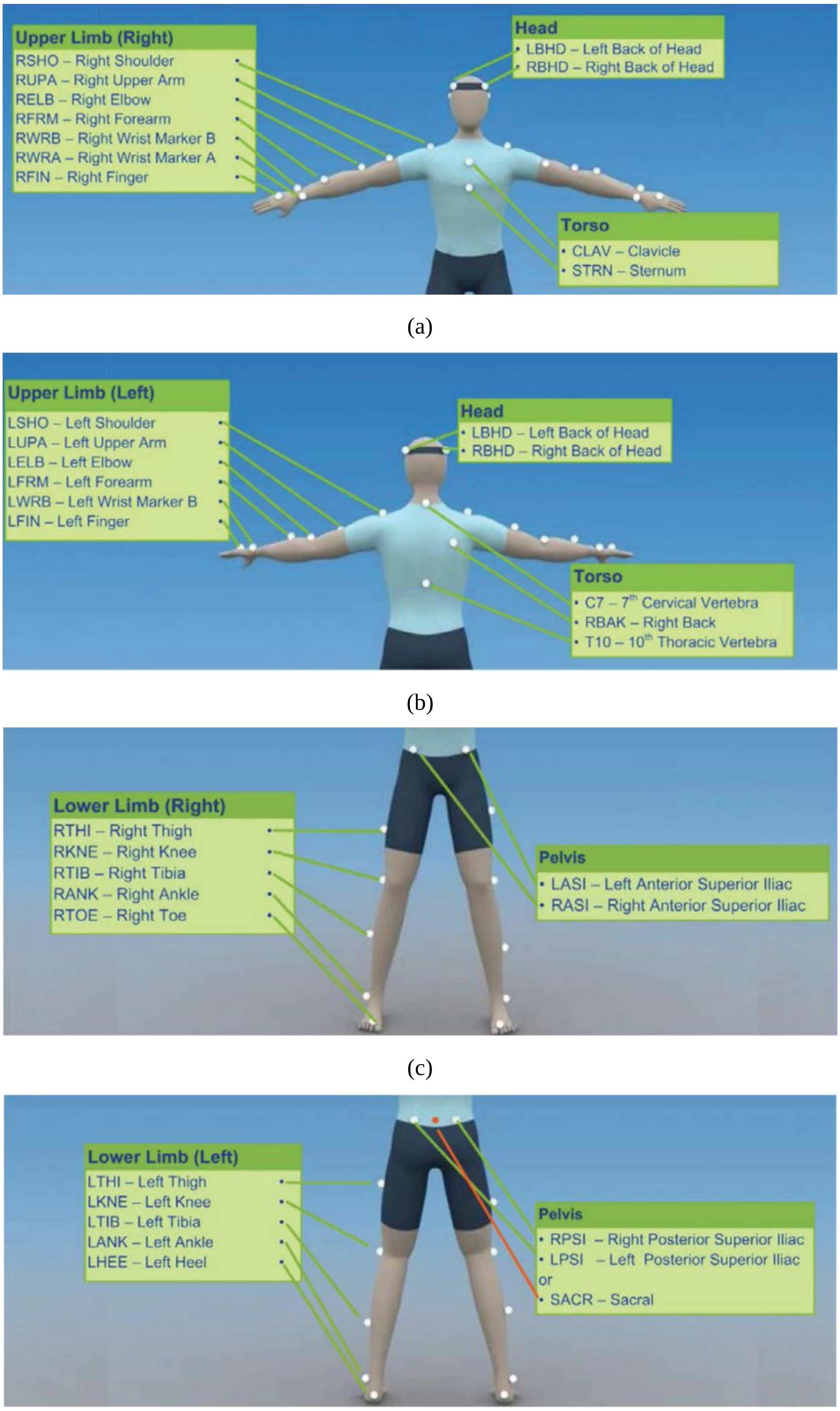
Movement	Description	Non-Optimal Movement
Deep squat	Subject bends the knees to descends the body toward the floor with the heels on the floor, the knees aligned over the feet, the upper body remains aligned in the vertical plane	Subject does not maintain upright trunk posture, unable to squat past parallel, demonstrates knee valgus collapse or trunk flexion greater than 30°
Standing shoulder abduction	Subject raises one arm to the side by a lateral rotation, keeping the elbow and wrist straight	Subject unable to maintain upright trunk posture or head in neutral position, lift arm does not remain in plane of motion, less than 160° of abduction



(a) (b)

Figure 3.1. Examples of the performed movements by one of the subjects: (a) Deep squat movement; (b) Standing shoulder abduction movement.

A Vicon optical tracking system was used for the data collection, which employs eight high-resolution cameras for tracking the position of 39 reflective markers attached to strategic locations on a subject's body. The locations for attaching the reflective markers [47] are shown in Figure 3.2. The optical tracking system captured the executed motions at 100 frames per second, while a dedicated software program assembled the recorded data into sequences of joint angle positions. The output data by the motion capture system are time-series consisting of 117–dimensional vectors of joint angle displacements. The order of measurements for the Vicon system is presented in Table 3.3. The joints for which the measurements are absolute are given with respect to the coordinate system of the sensory system and are indicated in the parenthesis in the table. For the remaining joints, the measurements are relative, and are given with respect to the parent joint in the skeletal model. The angle outputs for all joints are represented with the YXZ triplet of Euler angles in degrees.



(a)

(b)

(c)

(d)

Figure 3.2. Locations on the body for attaching the Vicon markers. (a) Front view of the upper body; (b) Back view of the upper body; (c) Front view of the lower body; (d) Back view of the lower body. The pictures are taken from [47]. Copyright: © 2016 Vicon Motion Systems Limited.

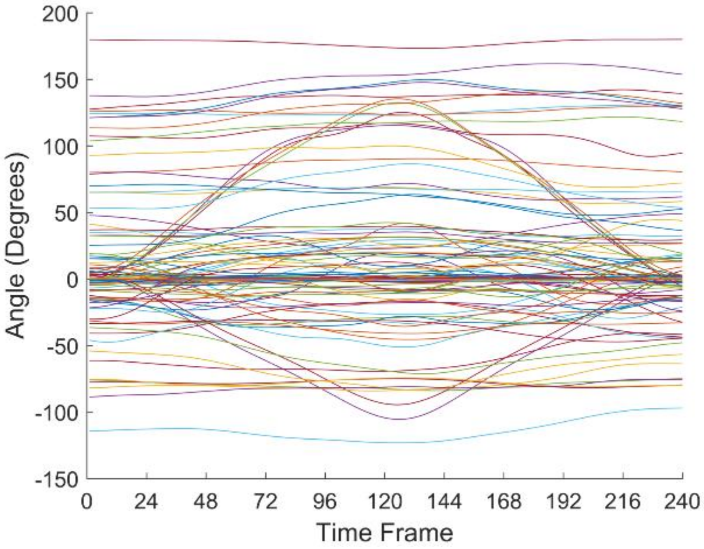
Table 3.3. Order of positions and angles in the data set for the Vicon optical tracker.

Joint order	Vicon Positions	Vicon Angles
1	LFHD - Left head front	Head (absolute)
2	RFHD - Right head front	Left head
3	LBHD – Left back head	Right head
4	RBHD - Right back head	Left neck
5	C7 - 7th cervical vertebra	Right neck
6	T10 - 10th thoracic vertebra	Left clavicle
7	CLAV - Clavicle	Right clavicle
8	STRN - Sternum	Thorax (absolute)
9	RBAK – Right back	Left thorax
10	LSHO - Left shoulder	Right thorax
11	LUPA - Left upper arm	Pelvis (absolute)
12	LELB - Left elbow	Left pelvis
13	LFRM - Left forearm	Right pelvis
14	LWRA - Left wrist A	Left hip
15	LWRB - Left wrist B	Right hip
16	LFIN - Left finger	Left femur
17	RSHO - Right shoulder	Right femur
18	RUPA - Right upper arm	Left knee
19	RELB - Right elbow	Right knee
20	RFRM - Right forearm	Left tibia
21	RWRA - Right wrist A	Right tibia
22	RWRB - Right wrist B	Left ankle
23	RFIN - Right finger	Right ankle
24	LASI - Left ASIS	Left foot
25	RASI - Right ASIS	Right foot

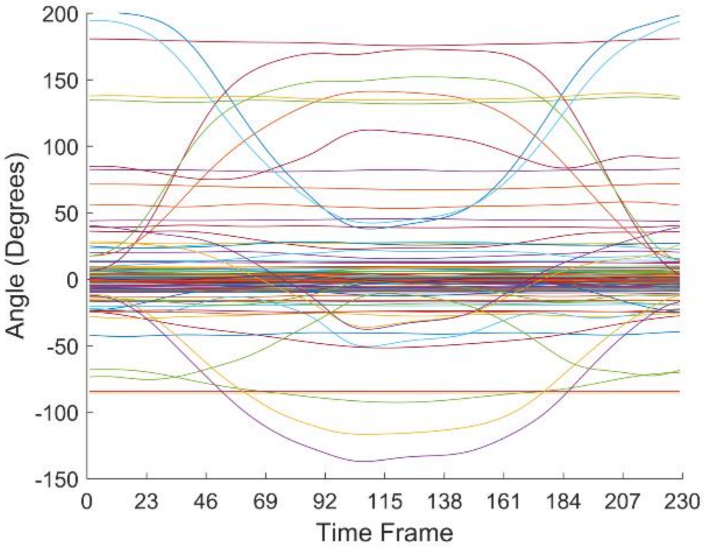
26	LPSI - Left PSIS	Left toe
27	RPSI - Right PSIS	Right toe
28	LTHI - Left thigh	Left shoulder
29	LKNE - Left knee	Right shoulder
30	LTIB - Left tibia	Left elbow
31	LANK - Left ankle	Right elbow
32	LHEE - Left heel	Left radius
33	LTOE - Left toe	Right radius
34	RTHI - Right thigh	Left wrist
35	RKNE - Right knee	Right wrist
36	RTIB - Right tibia	Left upper hand
37	RANK - Right ankle	Right upperhand
38	RHEE - Right heel	Left hand
39	RTOE - Right toe	Right hand

The single repetitions of each movement were separated, by identifying the beginning and end time steps of each repetition. Consequently, this resulted in a dataset consisting of 100 instances of correctly performed repetitions, and 100 instances of incorrectly performed repetitions, for each movement. By elimination of poorly recorded repetitions, as well as elimination of the data of subjects who performed the standing shoulder abduction exercise with their left arm (versus the rest of the subject who used their right arm), the final number of repetitions was reduced to 90 samples for Movement 1, and 63 samples for Movement 2. The number of correct and incorrect repetitions was kept equivalent for the two movements.

The angular movements of a single sequence for each exercise after the segmentation are shown in Figure 3.3. One can note that a majority of the 117 dimensions for both movements exhibit little to no variation throughout the exercise. This is more noticeable in the standing shoulder abduction in Figure 3.3(b), as the exercise involves only the movement of one of the subject's arms. As a result, both movements can be represented by only a few of the 117 dimensions, and therefore modeling of said movements can be achieved by the extraction and evaluation of these key components through dimensionality reduction. The corresponding length of the sequences for the two movements is 240 and 230 time steps, respectively.



(a)



(b)

Figure 3.3. Single sequence representation of all 117 dimensions for: (a) Deep squat movement; (b) Standing shoulder abduction movement.

3.2 Data Notation

The number of repetitions of a movement is denoted *N*, and the sequence of measurements by the optical tracking system for each correctly performed repetition is denoted $\mathbf{U}_n$, where *n* is used to index the individual sequences. The set of correct repetitions of a movement forms $\mathcal{U}=\{\mathbf{U}_n\}_{n=1}^{N}$. Each sequence $\mathbf{U}_n$ contains M temporally ordered vectors $\mathbf{U}_n=\left(\mathbf{u}_n^{(1)},\mathbf{u}_n^{(2)},\ldots,\mathbf{u}_n^{(M)}\right)$, where each temporal measurement is a *D*-dimensional vector, i.e., $\mathbf{u}_n^{(m)}\in\mathbb{R}^D$. The adopted notation employs bold fonts for vectors and matrices.

Similarly, the set of incorrect repetitions of the movements is denoted $\mathcal{W}=\{\mathbf{W}_n\}_{n=1}^{N}$. Each movement sequence $\mathbf{W}_n$ consists of *M* vectors $\mathbf{w}_n^{(m)}\in\mathbb{R}^D$, for $m=1,\,2,\ldots,M$.

3.3 Data Preprocessing and Labeling

The data preprocessing included scaling of the angular displacement measurements in the range $[-1,+1]$. More specifically, all sequences in the correct and incorrect movement sets were divided by the maximum absolute value of the correct set, i.e., $\max\left(\left|u_n^{(m)}\right|\right)$ for $n=1,\,2,\ldots,N$, $m=1,\,2,\ldots,M$. In addition, each movement sequence $\mathbf{U}_n$ and $\mathbf{W}_n$ was zero-mean shifted. Although it is commonly recommended to normalize the inputs to NNs into data vectors with a variance of 1, this is not applicable to the movement data since the variability of the individual dimensions is an important attribute of the data and needs to be preserved. Finally, the movement sequences for both exercises were aligned utilizing a temporal linear alignment method based on cubic interpolation of the data points. This was accomplished by determining the mean sequence length for each exercise and applying the mean sequence length to all remaining sequences.

As one of the goals of the considered task is to evaluate the level of correctness in the execution of movement repetitions during rehabilitation exercises, soft labels are assigned to each repetition instance. Root-mean-squared (RMS) deviation was adopted here as a metric for assessment of the repetition consistency. For this purpose, the RMS distance between each correct sequence $\mathbf{U}_n$ and the entire set $\mathcal{U}$ is calculated, i.e.,

$$\xi_i = \frac{1}{N}\sum_{n=1}^{N}\sqrt{\frac{1}{M}\sum_{i=1}^{M}\left(\mathbf{u}_n^{(i)} - \mathbf{u}_n^{(m)}\right)^2} \text{, for } i = 1, 2, ..., N. \tag{5}$$

Similarly, the RMS distance between each incorrect repetition $\mathbf{W}_n$ and the set of correct movements $\mathcal{U}$ is calculated as

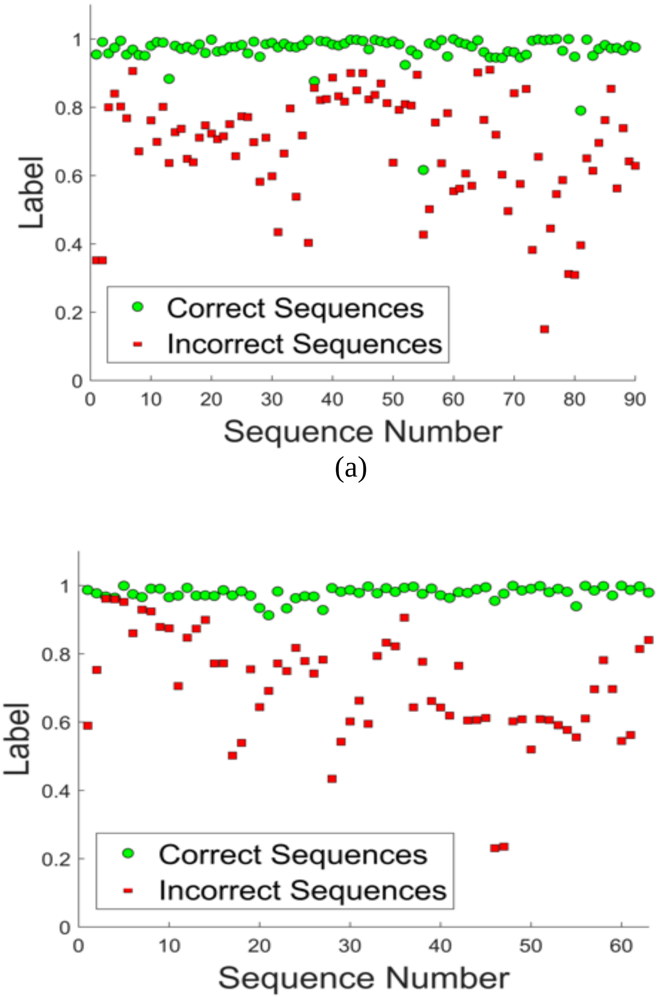
$$\zeta_i = \frac{1}{N}\sum_{n=1}^{N}\sqrt{\frac{1}{M}\sum_{i=1}^{M}\left(\mathbf{w}_n^{(i)} - \mathbf{u}_n^{(m)}\right)^2} \text{, for } i = 1, 2, ..., N. \tag{6}$$

One can note that in Eq. (6) the RMS deviation is calculated with respect to the set of correct movements.

Soft labels are assigned next to each of the correct and incorrect data sequences as follows:

$$l_i = \frac{1-\xi_i-\overline{\xi}}{\tau},\; l_i = \frac{1-\zeta_i-\overline{\xi}}{\tau} \text{, for } i = 1, 2, ..., N, \tag{7}$$

The resulting soft labels for the two movements are shown in Figure 3.4. In Eq. (7), $\overline{\xi}$ denotes the average value of the set of distances ξ. The parameter τ in Eq. (7) is a normalization factor that was empirically assigned the value of 100 for Movement 1 and 200 for Movement 2. The labels in Eq. (7) were set with a goal to be distributed in the range $[0, +1]$, and to retain a separation boundary between the correct and incorrect movements. It can be noticed in Figure 3.4 that several of the correct movements are performed in an inconsistent manner, and they are less similar to the remaining correct set of movements than some of the incorrectly performed movements. That was one motivation to introduce soft labels for the movement instances, instead of employing hard labels of 1's for the correct movements and 0's for the incorrect movements.



(a)

(b)

Figure 3.4. Soft labels for: (a) Deep squat movement; (b) Standing shoulder abduction movement. The labels for both correct and incorrect sequences for the movements are shown in the figure.

Furthermore, as stated earlier, one of our objectives is to assess the potential of GANs for evaluation of the level of correctness of therapy movements. The provision of soft labels allows to train an NN on a set of correct and incorrect movements, and to validate the trained networks on another set of correct and incorrect movements. Also, with the use of soft labels, the problem was cast from binary classification into a one-class classification, where all data instances belong to the same class of movement but have varying levels of movement quality. Additionally, we believe that the use of soft labels provides richer information of the input data and a basis for improved performance of both the generator and discriminator subnetworks.

One final note regarding the above procedure for applying soft labels to the motion data is that RMS deviation is probably a suboptimal metric for quantifying the distance between the high-dimensional data sequences. Although it was adopted here for proof of concept, the selection of metrics for the task at hand is one of the authors' topics for future research.

CHAPTER 4: EXPERIMENTAL RESULTS

4.1 Network Architectures

The thesis investigates the GAN variations presented in Chapter 2 (and their sub-variants in one case). A basis for comparison of the considered architectures is the DCGAN model depicted in Figure 4.1. The generative subnetwork consists of one fully connected layer and three padded convolutional layers. Following the guidelines in the DCGAN paper [37], ReLU activation functions are used in the generator except in the last layer that uses Tanh activation, and strided convolutions are utilized instead of pooling layers. As illustrated in Figure 4.1, the discriminative subnetwork has three padded convolutional layers. Leaky ReLU activation functions are introduced in the discriminator, and a dropout rate of 20% was applied to prevent overfitting. Adam optimizer was the choice in both subnetworks.

The investigated GAN models are fully described in Table 4.1. The structures of the networks are based on the DCGAN model presented in Figure 4.1. The networks are explained in more detail in the next section.

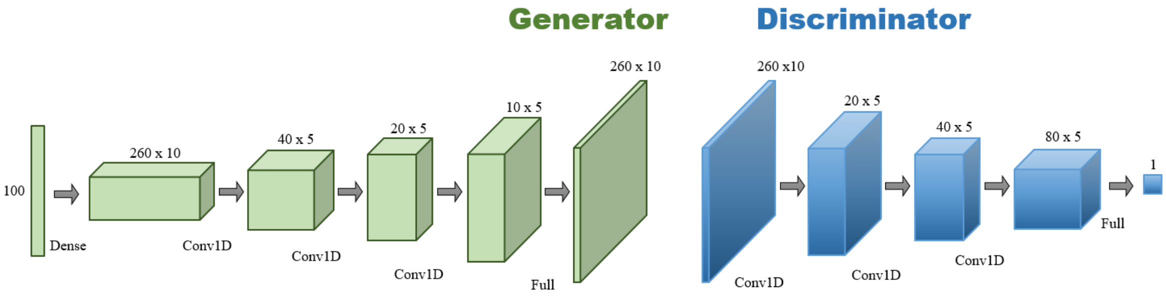


Figure 4.1. DCGAN model layers consisting of a generator and discriminator subnetworks composed of convolutional and MLP layers of hidden computational units.

Table 4.1. GANs network architectures with descriptions of generator and discriminator layers. Acronyms: LR – Leaky ReLU activation, R – ReLU activation, TH – Tanh activation, S – Sigmoid activation, BN – Batch normalization, US – Upsampling, D – Dropout, St – Strides, SGD – Stochastic Gradient Descent.

Network	**Generator**	**Discriminator**
GAN	50 (LR) × 100 (LR) × 200 (LR) × *M* = 260, *D* = 10 (TH): Adam	100 (LR,D) × 50 (LR,D) × 1 (S): Adam
DCGAN-1	100 (R, BN) × *M* = 260, *D* = 10 (R, BN) × Conv1D (40, 5, R, BN) × US(2) × Conv1D (20, 5, R, BN) × US(2) × Conv1D (*D* = 10, 5, TH): Adam	Conv1D (20, 5, LR, D, St:2) × Conv1D (40, 5, LR, D, BN) × Conv1D (80, 5, LR,D, BN) × 1 (S): Adam
DCGAN-2	100 (LR, BN) × *M* =260, *D* = 10 (LR) × Conv1D (40, 5, LR) × US(2) × Conv1D (20, 5, TH) × US(2) × Conv1D (*D* = 10, 5, TH): Adam	Conv1D (10, 5, LR, D, St:2) × Conv1D (20, 5, LR, D) × Conv1D (40, 5, LR,D) × 50 (LR,D) × 1 (S): Adam
WGAN	100 (LR) × *M* = 260, *D* = 10 (LR) × Conv1D (40, 5, LR) × US(2) × Conv1D (20, 5, LR) × US(2) × Conv1D (*D* = 10, 5, TH): Adam	Conv1D (10, 5, LR, D, St:2) × Conv1D (20, 5, LR, D) × Conv1D (40, 5, LR,D) × 50 (LR,D) × 1 (S): SGD
RGAN	(*M* = 260,5) × LSTM(100) : Adam	LSTM(100) × 1 (S): SGD

4.2 Movement Generation

The performance of the GAN representations listed in Table 4.1 is examined in relation to their capacity to generate data samples that resemble the time-series data of the actual physical therapy movements.

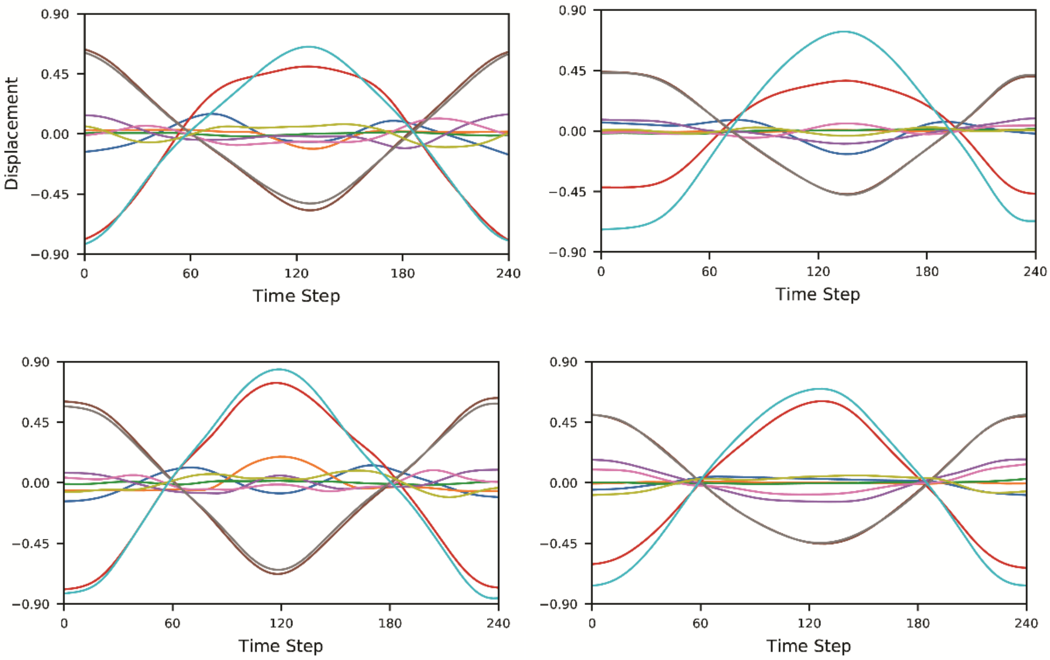
A subset of the data with reduced dimensionality is first considered, where 10 dimensions with the largest variation are extracted and used as input to the network. Several examples of the sequences for Movement 1 are presented in Figure 4.2(a).

One undesirable effect in the synthetic data samples produced by the GAN models is the distortion of the ends and beginnings of the generated sequences. To reduce the effect of the distortions, 10 time steps of synthetic data were added at the beginning and at the end of each sequence. The beginning 10 time steps were set equal to the first vector in each sequence, and the ending 10 time steps were set equal to the last vector in the sequence. Consequently, for Movement 1 the number of time steps M was increased from 240 to 260, and for Movement 2 the length M was increased from 233 to 250 time steps.

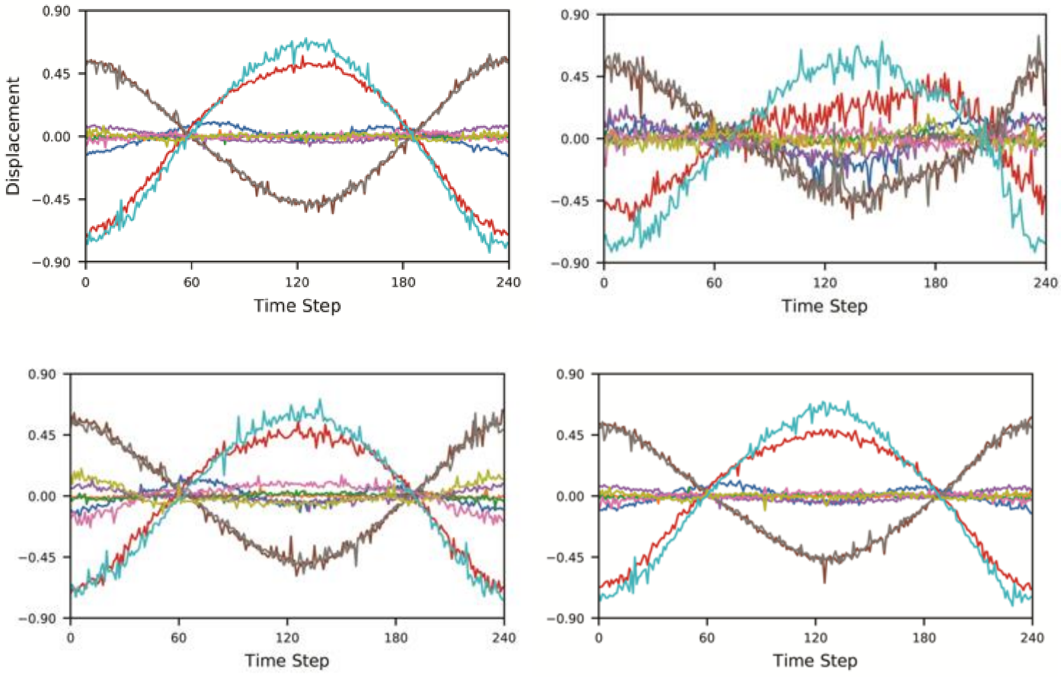
The GAN architectures in Table 4.1 are related to processing the input data for Movement 1, with the number of time steps M = 260, and dimensionality D = 10. The NNs for Movement 2 and the presented cases with different dimensionality have the same structure as the GANs presented in Table 4.1, and only the parameters M and D are varied.

For Movement 1, the subset for training purposes includes 70 correct and 70 incorrect movement repetitions, and the validation subset consists of the remaining 20 correct and 20 incorrect sequences. Similarly, for Movement 2, the training and validation subsets have 98 and 28 sequences of correct and incorrect repetitions, respectively.

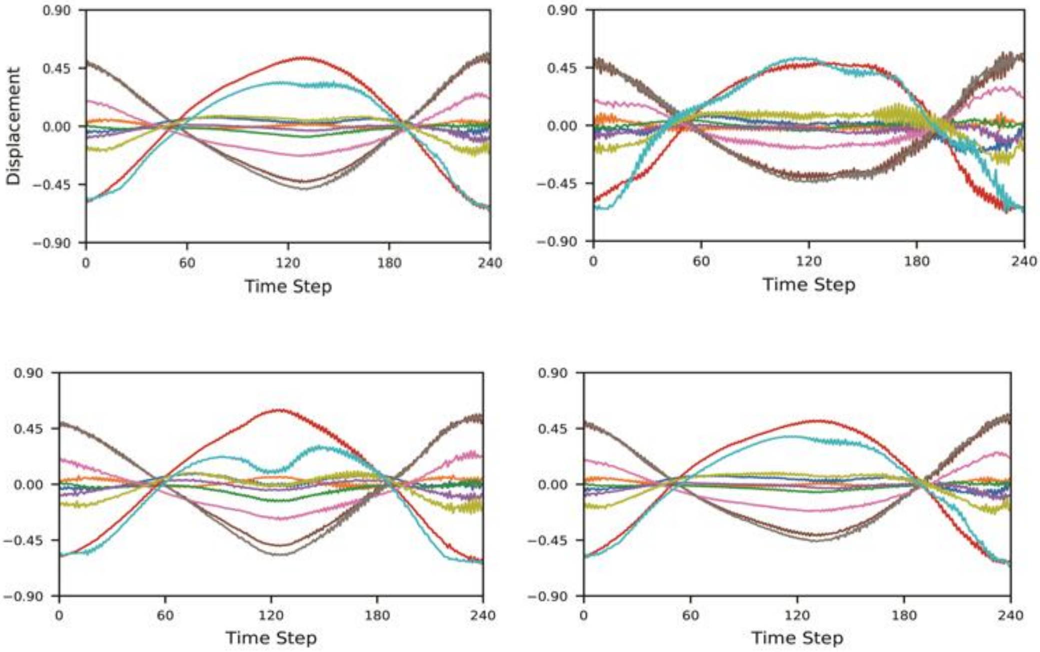
The sequences generated with the original GAN model [32] based on the structure outlined in Table 4.1 and consisting of MLP layers of computational units are shown in Figure 4.2(b). Conclusively, the data is quite noisy, and the network experiences a mode collapse early in the training, failing to refine the output of the generator. The next examined model is DCGAN-1 from Table 4.1, which implements the network structure recommended by the authors in reference [37]. However, the model was not able to produce data that resemble the real motion sequences. One potential reason is that the DCGAN network design reported in [37] is more applicable to image data. The suggested batch normalization of the hidden layers was the main contributing factor for the network failure with the human movement input data. Nevertheless, a variant of the model listed in Table 4.1 as DCGAN-2 provided realistic synthetic data. This network employs convolutional layers in a slightly altered architecture in comparison to the recommended DCGAN-1 model. Several representative examples of the generated sequences by DCGAN-2 are shown in Figure 4.2(c). Next, instances of the synthetic data generated with WGAN [38] are shown in Figure 4.2(d). The quality of the data is comparable to the sequences generated with DCGAN-2. Overall, WGAN model exhibited improved stability during training and, to a certain extent, visually improved the quality of generated data. The last investigated model is RGAN [36] with the network structure presented in Table 4.1, consisting of recurrent LSTM computational units. A set of generated data is displayed in Figure 4.2(e). The RGAN model created the smoothest synthetic sequences for Movement 1, and it outperformed the other models that are based on convolutional and MLP layers of hidden units.



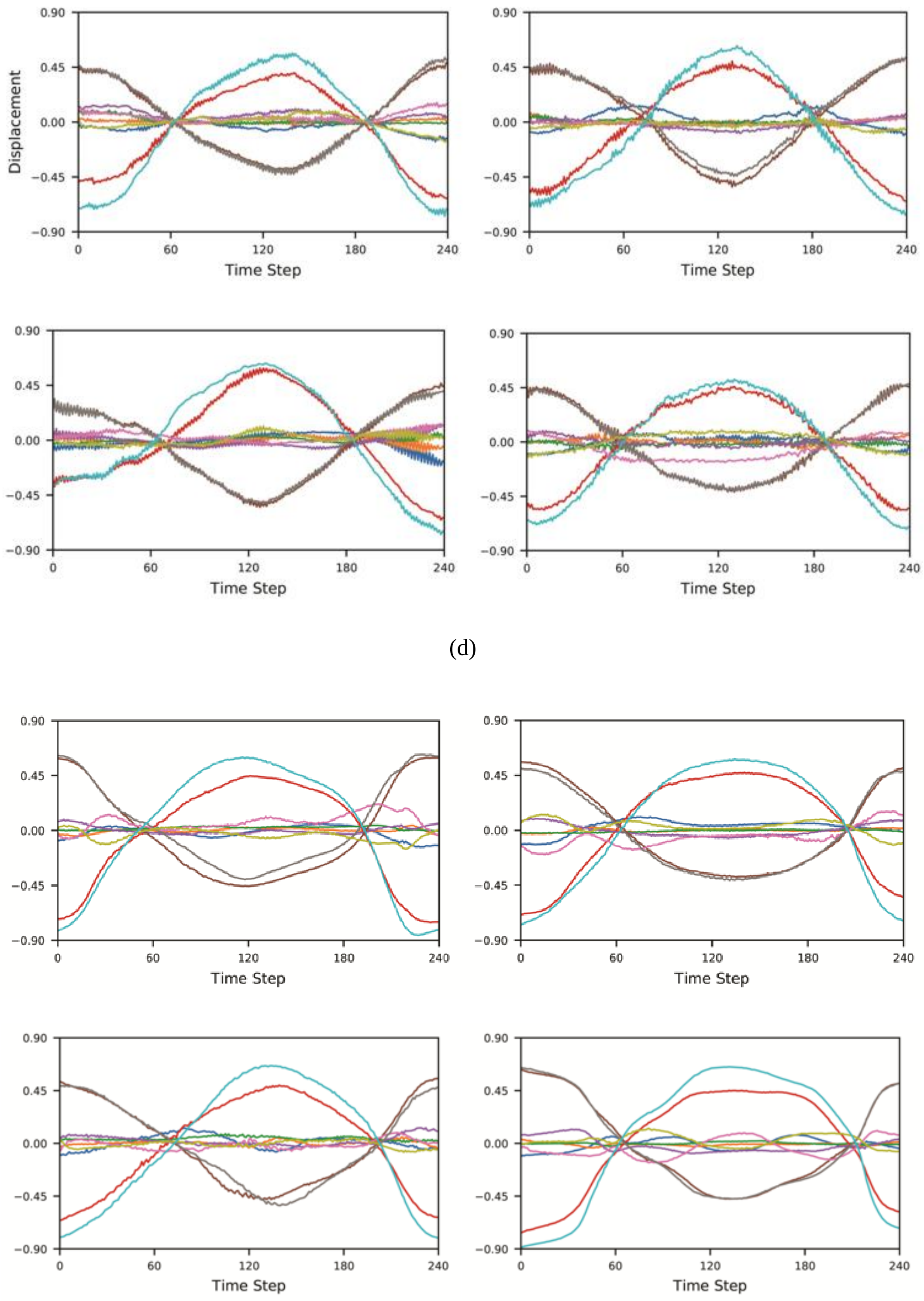
(a)


Displacement
0.90
0.45
0.00
−0.45
−0.90
0
60
120
180
240
Time Step

(b)


Displacement
0.90
0.45
0.00
−0.45
−0.90
0
60
120
180
240
Time Step

(c)



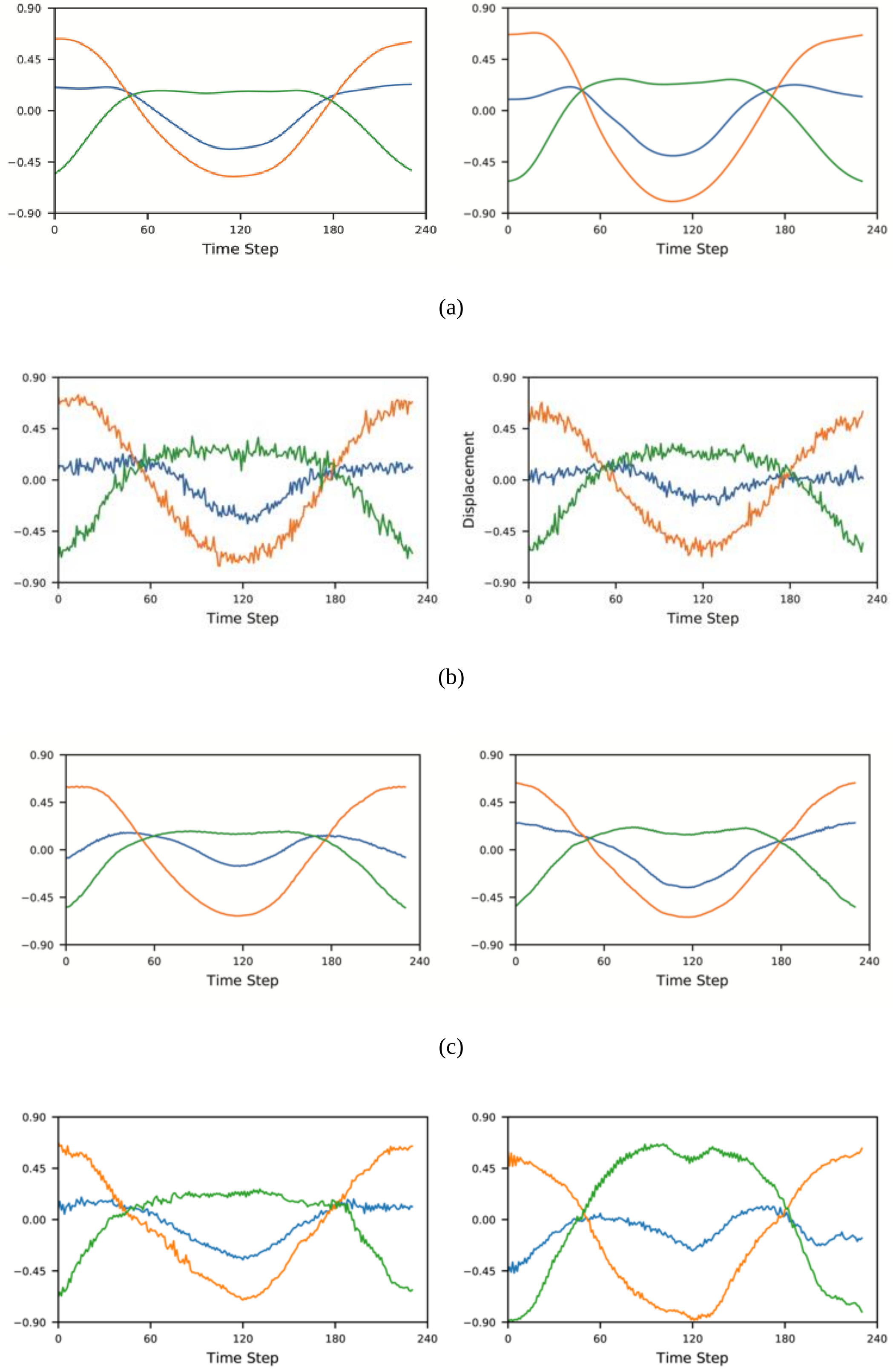
(d)

(e)

Figure 4.2. (a) Samples of 10-dimensional Movement 1 sequences as recorded with the optical tracking system. (b) Examples of generated sequences with the GAN network from Table 4.1. (c) Examples of generated sequences with the DCGAN-2 network from Table 4.1. (d) Examples of generated sequences with the WGAN network from Table 4.1. (e) Examples of generated sequences with the RGAN network from Table 4.1.

Another validation case is presented next for Movement 2, related to the standing shoulder abduction exercise. In this case, the time-series dimensionality is reduced to the three dimensions with the largest variance. Considering the strong correlation between the joint angular displacements in human movements, a body of work in the literature relied on only several most important dimensions for motion modeling. As expected, for the considered exercise, the dimensions with the largest variability correspond to the angular displacements of the upper arm, lower hand, and the wrist. Two movement repetitions as acquired by the optical tracker are displayed in Figure 4.3(a). Similar to the first validation case, the networks presented in Table 4.1 are employed for modeling the movements and generating synthetic data samples. Instances of the generated sequences with the conventional GAN model are shown in Figure 4.3(b), and similar to Figure 4.2(b), the sequences are quite noisy. Examples of the generated data with the DCGAN-2 and WGAN models are shown in Figures 8(c) and (d), respectively. The quality of the GAN-generated sequences is visually appealing, and one can notice that the networks demonstrated improved performance in the case of low-dimensional input data. Conversely, the samples generated with DCGAN-2 are less smooth for this movement. The generated data with RGAN are presented in Figure 4.3(e).

In summary, the RGAN model produced the smoothest and visually attractive synthetic sequences for the two movements. The GAN models based on layers of convolutional kernels were also able to generate data sequences of comparable and acceptable quality. The synthetic data samples produced with the original GAN model are the least smooth when compared to the other cases, although the model was able to learn the general pattern of the movement sequences.

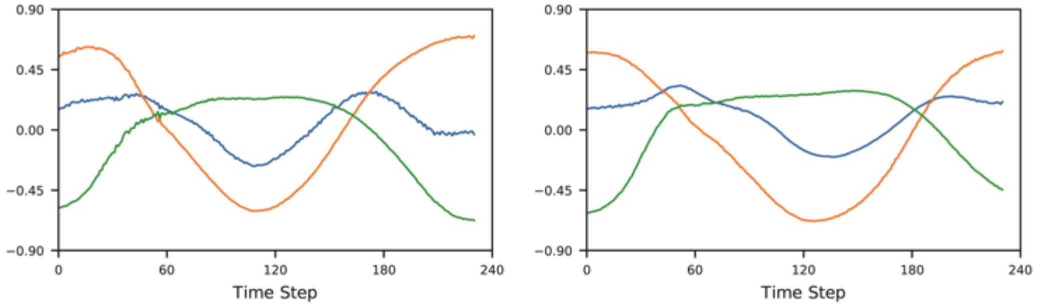


(a)

(b)

(c)

(d)



(e)

Figure 4.3. (a) Samples of 3-dimensional Movement 2 sequences as recorded with the optical tracking system. (b) Examples of generated sequences with the GAN network from Table 4.1. (c) Examples of generated sequences with the DCGAN-2 network from Table 4.1. (d) Examples of generated sequences with the WGAN network from Table 4.1. (e) Examples of generated sequences with the RGAN network from Table 4.1.

4.3 Movement Classification

Next, the ability of the GANs presented in Table 4.1 to classify therapy movement repetitions is evaluated. For comparing the performance of the models, a metric is adopted which sums the absolute differences between the predicted probabilities of the discriminator and the soft labels for the data instances $\mathbf{X}_k$ in the validation subset l_k, i.e.,

$$C=\sum_{k=1}^{K}\left|\mathcal{P}\left(\mathbf{X}_k\right)-l_k\right|, \tag{8}$$

where K denotes the number of validation sequences.

The values of the metric C for the considered GAN models are presented in Table 4.2. Presented in the table also are the performance scores of NNs consisting only of the discriminator subnetwork (i.e., without a generator subnetwork). In Table 4.2, the corresponding NNs have an extension “-Disc.” The values of the metric for the WGAN model are not presented in the table, as the outputs of its discriminator are not probabilities (but instead are values of the Wasserstein distance). Table 4.2 contains the distances C for cases of 3-dimensional (3D) and 10-dimensional (10M) movement sequences.

For the discriminative NNs in Table 4.2, the presented numbers correspond to the average value of the parameter *C* based on five runs of the models. The values in the parenthesis preceded with the symbol S are the respective standard deviations. Early stopping of 100 epochs was employed in the training phase.

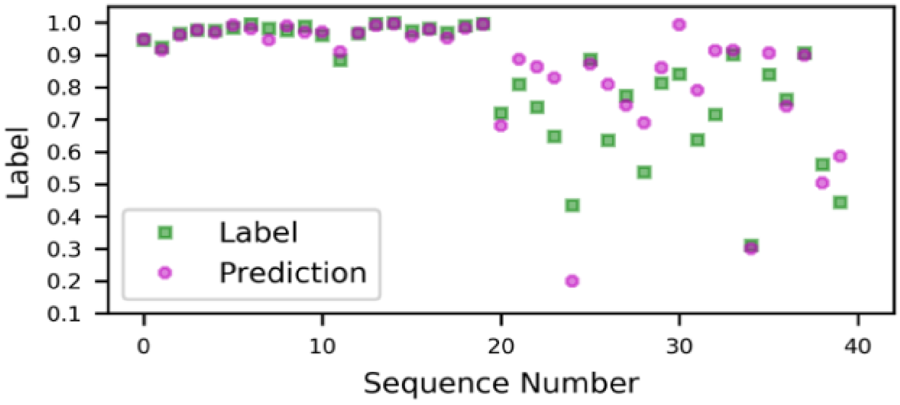
For the GAN models, the presented values of the parameter *C* in Table 4.2 are based on a single run of the networks. In particular, the values in the parenthesis preceded with the symbol M are the minimum values of the parameter *C*, whereas the upper numbers represent the average values of the parameter *C* based on the preceding 25 epochs and the succeeding 25 epochs relative to the minimum value. Averaging was employed in order to filter out the significant oscillations in the obtained *C* values with the GAN models.

Table 4.2. Classification accuracy results for GANs and the corresponding discriminative models. Notation: M – minimum value; S – standard deviation; Disc – a discriminator model only without a generator.

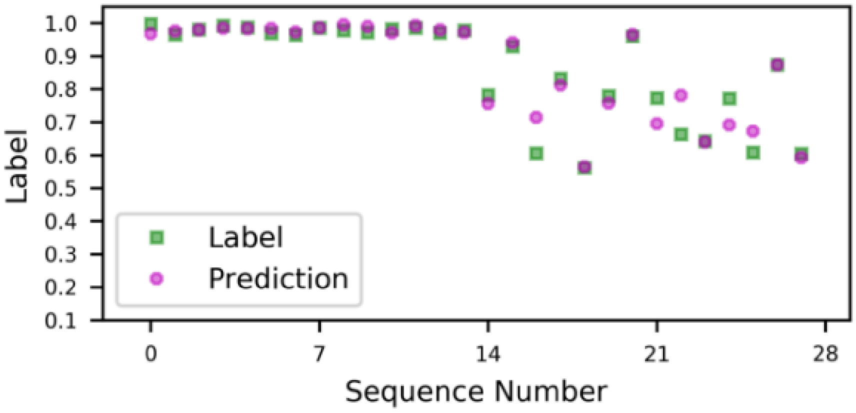
Network	**Movement 1**		**Movement 2**	
	3-dimensional	**10-dimensional**	**3-dimensional**	**10-dimensional**
GAN	2.220 (**M1.82**)	2.097 (**M1.79**)	0.801 (**M0.58**)	0.797 (**M0.60**)
GAN-Disc	2.254 (S±0.05)	2.683 (S±0.14)	1.008 (S±0.10)	0.922 (S±0.04)
DCGAN-1	3.965 (**M2.60**)	2.237 (**M2.00**)	1.136 (M0.98)	0.789 (**M0.61**)
DCGAN-1-Disc	3.251 (S±0.63)	2.413 (S±0.05)	0.866 (S±0.02)	0.852 (S±0.22)
DCGAN-2	3.649 (**M1.86**)	1.999 (**M1.33**)	0.836 (**M0.74**)	0.793 (**M0.64**)
DCGAN-2-Disc	2.309 (S±0.16)	2.057 (S±0.31)	0.799 (S±0.01)	0.947 (S±0.00)
RGAN-Disc	2.637 (S±0.16)	2.446 (S±0.45)	1.336 (S±0.14)	0.878 (S±0.04)

One example of the performance of the considered models is depicted in Figure 4.4. The figure shows the soft labels calculated based on Eq. (7) and the output probabilities of the DCGAN-1-Disc model. Figure 4.4(a) displays the scores for Movement 1, which has a validation set of 40 sequences. In the figure, the first 20 sequences are drawn from the set of correct movements, and the last 20 sequences are drawn from the set of incorrect movements. One can notice that the network evaluates the correct movements very accurately, and that for the incorrect movements the network predictions are close to the assigned labels. Similarly, Figure 4.4(b) presents the labels and the network predictions for Movement 2, for which the validation set consists of 28 data sequences. The predicted labels for the movement repetitions for this case also approximate the actual labels.

From the results in Table 4.2 regarding the discriminative NNs, it can be concluded that DCGAN-2-Disc achieved the lowest cumulative deviation between the input soft labels and the predicted labels, in comparison to the other discriminative models. Overall, the 10-dimensional sequences provided richer discriminative information of the movements and produced better results in comparison to the 3-dimensional sequences. The discriminators of the original GAN and DCGAN-1 also achieved comparable classification accuracy.



(a)



(b)

Figure 4.4. Soft labels and predicted labels by the DCGAN-1-Disc model for: (a) Deep squat movement; (b) Standing shoulder abduction movement.

In comparison to the discriminative NNs, the predicted labels of the movements by the GAN architectures are characterized by lower deviation values *C* in relation to the input labels. The obtained values are shown with a bold font in Table 4.2. Almost in all cases, the GAN models outperformed the discriminative NNs. The discriminator based on recurrent computational units RGAN-Disc produced lower or comparable classification accuracies, compared to the models with convolutional units. The RGAN demonstrated lower classification accuracy and the results are not shown in the table.

Among the drawbacks of employing GANs for this task is the computational expense, as the GAN networks took significantly longer to train in comparison to the discriminative NNs, and in some cases, the GAN models required an additional fine-tuning of the hyperparameters to obtain the reported classification accuracy.

4.4 Movement Augmentation

Data augmentation is a fundamental technique for addressing the problem of limited data in machine learning. The objective is to boost the diversity of data for model training, which consequently, can reduce the possibility for overfitting, as well as it can improve model robustness. Unlike most image processing applications that take advantage of large-scale open datasets of annotated general-purpose images, biomedical human movement applications do not have access to large datasets, and therefore, data augmentation is indispensable for processing movement data with machine learning models. In image processing applications, data augmentation is commonly implemented by applying image translation and rotation, and various other image operations, such as applying a small amount of random noise to images, rescaling the pixels' intensity, adjusting the gamma value of image brightness, adjusting the sigmoid value of image contrast, and similar. Differently, time-series data are sparser than images, and thus, there are reduced opportunities for augmentation in comparison to image data.

This section investigates augmentation of the recorded sets of rehabilitation movement sequences by adding a small amount of noise to the time-series data. For a movement sequence $\mathbf{X}_n$ (which can represent either correct or incorrect repetitions of a movement), new synthetic sequences are generated by

$$\hat{\mathbf{X}}_n = \mathbf{X}_n + q \cdot \mathbf{v} \text{ for } n = 1, 2, ..., N, \tag{9}$$

where $\mathbf{v} \sim V(0,1)$ denotes a sequence of random numbers sampled from a uniform probability distribution V and with the same dimensionality as the sequence $\mathbf{X}_n$, and q is a constant with a value that is varied within a range, as explained in the subsequent text.

Three deep learning models are adopted for data augmentation evaluation, hereafter referred to as CNN, RNN, and HNN. CNN is an architecture containing convolutional hidden layers, RNN is based on recurrent hidden layers, and HNN has a hierarchical network structure. For the three models, we conducted a grid search to finetune the hyperparameters consisting of combinations of NN layers, numbers of layers, nodes per layer, filter size, and batch size. The resulting NNs architectures are presented in Table 4.3.

The resulting CNN model has three convolutional layers, two fully connected layers, and an output layer with linear activations. The convolutional layers contain strided 1D convolutional filters, leaky ReLU activation functions, and a dropout rate of 0.2.

The RNN model contains two bidirectional layers of LSTM units, a fully connected layer, and an output layer. The fully connected layer contains Leaky ReLU activation functions with a dropout rate of 0.5. The recurrent layers apply a recurrent dropout of 0.5 and are followed by a dropout layer with 0.25 dropout rate.

Table 4.3. Architectures of CNN, RNN, and HNN models.

Acronyms: Conv1D (N_{K}, N_{S},…) – Layer with one-dimensional convolutional units with N_{K} kernels of size N_{S}, FC (…) – Fully connected layer, BiLSTM (…) – Layer with bidirectional LSTM units, BiRNN – Layer with bidirectional simple recurrent units, LR – Leaky ReLU activation, D – Dropout, RD – Recurrent dropout, St – Stride, L – linear activation, TH – Tanh activation, → Merged layers.

Networks	**Layers**
CNN	Conv1D (60, 5, LR, D:0.2, St:2) × Conv1D (30, 3, LR, D:0.2, St:2) × Conv1D (10, 3, LR, D:0.2) × FC (200, LR, D:0.2) × FC (100, LR, D:0.2) × FC (1, L) : Adam
RNN	BiLSTM (20, RD:0.5, D:0.25) × FC (30, LR, D:0.5) × BiLSTM (10, RD:0.5, D:0.25) × FC (1, L) : Adam
HNN	**{**BiRNN (10, TH, RD:0.5) * 5} → × **{**BiRNN (20, TH, RD:0.5) * 4} → × **{**BiRNN (20, TH, RD:0.5) * 2} → × BiLSTM (30, TH, RD:0.5) × FC (1, L) : Adam

The HNN model [19] has a hierarchical structure that contains five sub-networks with RNN layers. The five sub-networks take as inputs joint displacement data of the left arm, right arm, left leg, right leg, and torso, respectively. Such a hierarchical structure of HNN has the advantage of enabling low-level spatial information from joint coordinates to be leveraged for obtaining a high-level representation of the body parts’ movements. The model consists of three bidirectional layers with recurrent units, a bidirectional layer with LSTM units, and an output layer. The recurrent dropouts are 0.5, and the activation functions are Tanh in the bidirectional layers, and linear activations in the output layer.

The effect of data augmentation for the above three NNs was explored for the deep squat movement. Four different values for the parameter q in Eq. (9) were selected, i.e., $q \in \{0.01, 0.03, 0.05, 0.07\}$, that add different intensities of noise to the original data. By adding random noise to the instances of the deep squat exercise, additional instances were artificially generated, which resulted in a four-fold increase of the dataset. The NNs were trained using the original dataset and augmented dataset containing both the original and synthetic data. For training, we used mean-squared-error for minimizing the loss function, and Adam optimizer was selected for the parameters update. The batch size was set to 5, and early stopping regularization was applied to prevent overfitting. The inputs to the models were 117-dimensional sequences of joint displacements representing a single repetition of the deep squat movement. A linear activation function was applied to the output layer to regress a numerical value representing a movement quality score for an input repetition. The results are summarized in Table 4.4, indicating that for all three NNs the average absolute deviation dropped more than 50% when using the augmented datasets in comparison to the training only with the original data.

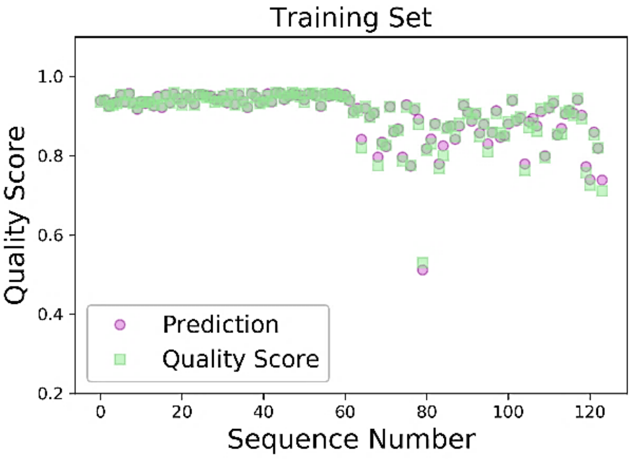
Table 4.4. Average absolute deviation for the deep squat exercise.

Data type	**CNN**	**RNN**	**HNN**
Original data	0.01357	0.01670	0.03010
Augmented data	**0.00656**	**0.00688**	**0.01404**

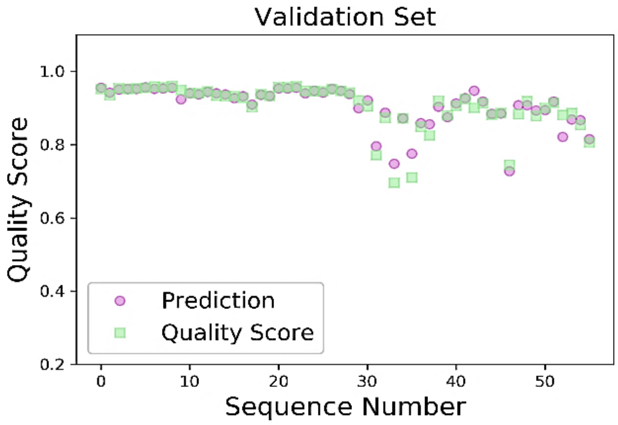
Predicted outputs by the CNN model for the deep squat exercise are depicted in Figure 4.5. The input data includes 90 correct and 90 incorrect repetitions. A random set of 124 of the repetitions was used for training and the remaining 56 were used for validation. To obtain movement quality scores for the individual repetitions, the set of correct sequences was first modeled with a Gaussian mixture model. Afterward, movement scores were calculated by using the likelihood that individual repetition data were drawn from the Gaussian mixture model as a performance metric [48]. The values of the likelihood were mapped to normalized values in the range [0, 1].

The input quality scores and the predictions for the training and validation sets are shown in Figure 4.5(a) and (b), respectively. The green squares in the figures represent ground truth scores from the Gaussian mixture model, and the red circles symbolize the predictions by the CNN model. The first half in both figures shows the corrected repetitions (and as expected, they have higher quality scores with values close to 1) and the second half in both figures shows the incorrect repetitions (having lower quality scores). Based on the figure, it can be concluded that the model correctly predicted the quality scores for almost all repetitions with a small deviation from the ground truth values used for training.

This demonstrates the potential of deep learning models for assessment of the level of performance of patients enrolled in physical rehabilitation programs. Such models can be used for providing real-time feedback to patients enrolled in home-based rehabilitation. Also, the models can be used for assisting clinicians by providing objective movement quality scores during the evaluation of patient performance in a clinic-based environment.



(a)



(b)

Figure 4.5. (a) CNN predictions on the training set for exercise for deep squat movement; (b) CNN predictions on the validation set for the exercise.

CHAPTER 5: CONCLUSION

The thesis employs GANs for modeling and evaluation of physical rehabilitation movements and for generating synthetic movement sequences. Four GAN models are considered, which include: GAN, DCGAN, WGAN, and RGAN. We selected these four models due to their relevance to the considered problem, or because they are popular GAN variations used for modeling spatial or time-series data. The ability of the networks to generate data instances that resemble two sets of physical therapy movements is evaluated. Further, the classification accuracy of the GANs and the ability to predict the level of performance of the exercises is evaluated based on introduced soft labels for the movement sequences. The results demonstrate the capacity of the considered GAN models to learn the underlying structure of the movement sequences, and with that, to generate realistic synthetic movement data, and to predict the level of performance consistency on a set of unseen movement sequences. These capabilities furnish a potential for augmentation of datasets of therapy movements with synthetically generated samples for improved movement modeling, and for utilization in automated monitoring and evaluation of the level of correctness of patient movements in home-based therapy programs. Also, the provision of means for synthesizing realistic time-series data can benefit other related application areas. We envision a real-world application of the investigated deep learning models on movement data collected with a vision-depth motion capture sensor.

References

[1] K. Czarnecki, J. M. Thompson, R. Seime, Y. E. Geda, J. R. Duffy, and J. E. Ahlskog, "Functional movement disorders: Successful treatment with a physical therapy rehabilitation protocol," *Parkinsonism & Related Disorders*, vol. 18, pp. 247–251, March 2012.

[2] M. E. Morris, "Movement disorders in people with Parkinson disease: A model for physical therapy," *Physical Therapy*, vol. 80, no. 6, pp. 578–597, June 2000.

[3] I. K. Ho, K. R. Goldschneider, S. Kashikar-Zuck, U. Kotagal, C. Tessman, and B. Jones, "Healthcare utilization and indirect burden among families of pediatric patients with chronic pain," *Journal of Musculoskeletal Pain*, vol. 16, no. 3, pp. 155–164, 2008.

[4] S. R. Machlin, J. Chevan, W. W. Yu, and M. W. Zodet, "Determinants of utilization and expenditures for episodes of ambulatory physical therapy among adults," *Physical Therapy*, vol. 91, no. 7, pp. 1018–1029, 2011.

[5] S. K. Saxena, T. P. Ng, D. Yong, N. P. Fong, and K. Gerald, "Total direct cost, length of hospital stay, institutional discharges and their determinants from rehabilitation settings in stroke patients," *Acta Neurologica Scandinavica*, vol. 114, no. 5, pp. 307–314, 2006.

[6] S. F. Bassett, and H. Prapavessis, "Home-based physical therapy intervention with adherence-enhancing strategies versus clinic-based management for patients with ankle sprains," *Physical Therapy*, vol. 87, no. 9, pp. 1132–1143, September 2007.

[7] E. M. Sluijs, G. Kok, and J. van der Zee, "Correlates of exercise compliance in physical therapy," *Physical Therapy*, vol. 73, no. 11, pp. 771–782, December 1993.

[8] Jack, S. M. McLean, J. K. Moffett, and E. Gardiner, “Barriers to treatment adherence in physiotherapy outpatient clinics: a systematic review,” *Manual Therapy,* vol. 15, no. 3, pp. 220–228, 2010.

[9] K. K. Miller, R. E. Porter, E. DeBaun-Sprague, M. Van Puymbroeck, and A. A. Schmid, “Exercise after stroke: patient adherence and beliefs after discharge from rehabilitation,” *Topics in Stroke Rehabilitation,* vol. 24, no. 2, pp. 142–148, 2017.

[10] V. G. Duffy, “Digital human modeling: Applications in health, safety and ergonomics and risk management,” in *Proc. International Conference on Digital Human Modeling*, Los Angeles, USA, 2015.

[11] J. Valasek, K. Kirkpatrick, J. May, and J. Harris, “Intelligent motion video guidance for unmanned air system ground target surveillance,” *Journal of Aerospace Information Systems,* vol. 13, pp. 10–26, January 2016.

[12] A. Vakanski, I. Mantegh, A. Irish and F. Janabi-Sharifi, “Trajectory learning for robot programming by demonstration using hidden Markov model and dynamic time warping,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics),* vol. 42, no. 4, pp. 1039–1052, August 2012.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. International Conference on Neural Information Processing Systems (NIPS),* Lake Tahoe, USA, pp. 1106–1114, 2012.

[14] K. Fragkiadaki, S. Levine, P. Felsen and J. Malik, “Recurrent network models for human dynamics,” in *Proc. IEEE International Conference on Computer Vision (ICCV),* pp. 4346–4354, 2015.

[15] A. Vakanski, J. M. Ferguson, and S, Lee, “Mathematical modeling and evaluation of human motions in physical therapy using mixture density neural networks,” *Journal of Physiotherapy & Physical Rehabilitation,* vol. 1, no. 4, pp. 1-10, December 2016.

[16] S. Ji, W. Xu, M. Yang, and K. Yu, “3D convolutional neural networks for human action recognition,” I*EEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, January 2013.

[17] J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural Networks*, vol. 61, pp 85–117, January 2015.

[18] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016, ch. 10, pp. 367–415.

[19] Y. Du, W. Wang, and L. Wang, “Hierarchical recurrent neural network for skeleton based action recognition,” in Proc. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1110–1118, 2015.

[20] W. Zhu, C. Lan, J, Xing, W, Zeng, Y. Li, L, Shen, and X. Xie, “Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks,” in *Proc. of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI'16)*, pp. 3697–3703, March 2016.

[21] A. T. Özdemir, and B. Barshan, “Detecting falls with wearable sensors using machine learning techniques,” *Sensors*, vol. 14, pp. 10691–10708, June 2014.

[22] A. Jain, A. Singh, H. S. Koppula, S. Soh, and A. Saxena, “Recurrent neural networks for driver activity anticipation via sensory-fusion architecture,” in *Proc. 2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3118–3125, May 2016.

[23] J. F. S. Lin, and D. Kulić, “Online segmentation of human motion for automated rehabilitation exercise analysis,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 1, pp. 168–180, Jan 2014.

[24] I. Ar, and Y. S. Akgul, “A computerized recognition system for the home-based physiotherapy exercises using an RGBD camera,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 6, pp. 1160–1171, Nov 2014.

[25] L. V. Calderita, P. Bustos, C. Suárez Mejías, F. Fernández, and A. Bandera, “THERAPIST: Towards an autonomous socially interactive robot for motor and neurorehabilitation therapies for children,” in *Proc. 7th International Conference on Pervasive Computing Technologies for Healthcare and Workshops*, pp. 374–377, Venice 2013.

[26] S. C. Yeh, M. C. Huang, P. C. Wang, T, Y. Fang, M. C. Su. P. Y. Tsai, and A. Rizzo, “Machine learning-based assessment tool for imbalance and vestibular dysfunction with virtual reality rehabilitation system,” *Computer Methods and Programs in Biomedicine*, vol.116, pp. 311–318, October 2014.

[27] F. J. Badesa, R. Morales, N. G. Aracil, J. M. Sabater, A. Casals, and L. Zollo, “Auto-adaptive robot-aided therapy using machine learning techniques,” *Computer Methods and Programs in Biomedicine*, vol. 116, pp. 123–130, September 2014.

[28] L. D. LLedo, A. Bertomeu, J. Diez, F. J. Badesa, R. Morales, J. M. Sabater, and N. G. Aracil, “Auto-adaptative robot-aided therapy based in 3d virtual tasks controlled by a supervised and dynamic neuro-fuzzy system,” *International Journal of Artificial Intelligence and Interactive Multimedia*, vol. 3, no. 2, pp. 63–68, Mar 2015.

[29] M. Ortiz-Catalan, R. A. Gudmundsdottir, M. B. Kristoffersen, A. Z. Ehvarria, K. C. Winterberger, K. K. Ortiz, *et al*., “Phantom motor execution facilitated by machine learning and augmented reality as treatment for phantom limb pain: a single group, clinical trial in patients with chronic intractable phantom limb pain,” *The Lancet*, vol. 388, December 2016.

[30] D. Antón, A. Goñi, A. Illarramendi, J. J. Torres-Unda, and J. Seco, "KiReS: A Kinect-based telerehabilitation system," *2013 IEEE 15th International Conference on e-Health Networking, Applications and Services (Healthcom 2013)*, pp. 444–448, October 2013.

[31] R. Komatireddy, A. Chokshi, J. Basnett, M. Casale, D. Goble, and T. Shubert "Quality and quantity of rehabilitation exercises delivered by a 3-D motion controlled camera: A pilot study," *International Journal of Physical Medicine and Rehabilitation*, vol. 2, no. 4, August 2014.

[32] I. J. Goodfellow, J. P. Abadie, M. Mirza, B. Xu, D. W. Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. International Conference on Neural Information Processing Systems (NIPS)*, 2014.

[33] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 105–114, July 2017.

[34] G. Antipov, M. Baccouche, and J. L. Dugelay, "Face aging with conditional generative adversarial networks," in *Proc. IEEE International Conference on Image Processing (ICIP)*, pp. 2089–2093, September 2017.

[35] Y. Hong, U. Hwang, J. Yoo, and S. Yoon, "How generative adversarial nets and its variants work: An overview of GAN," *arXiv*:1711.05914v6 [cs.LG], 2018.

[36] S. L. Hyland, C. Esteban, and G. Ratsch, "Real-valued (medical) time-series generation with recurrent conditional GANs," arXis:1706.02633v2 [stat.ML], 2017.

[37] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv*:1511.06434v2 [cs.LG], 2016.

[38] M. Arjovsky, S. Chintala, and L Bottou, “Wasserstein generative adversarial networks,” in *Proc. International Conference on Machine Learning (ICML)*, 2017.

[39] D. Berthelot, T. Schumm, and L. Metz, “BEGAN: Boundary equilibrium generative adversarial networks,” *arXiv*:1703.10717, 2017.

[40] X. Mao, Q. Li, H. Xie, R. Lau, Z. Wang, and S. P. Smolley. “Least squares generative adversarial network,” *arXiv*:1611.04076, 2016.

[41] M. Mirza, and S. Osindero, “Conditional generative adversarial nets,” *arXiv*:1411.1784v1 [cs.LG], 2014.

[42] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, “InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets,” *arXiv*:1606.03657v1 [cs.LG], 2016.

[43] J. Donahue, P. Krähenbühl, and T. Darrell, “Adversarial Feature Learning.,” in 5th *International Conference on Learning Representations*, Toulon, France, April 24-26, 2017.

[44] A. Brock, J. Donahue, and K. Simonyan, “Large Scale GAN Training for High Fidelity Natural Image Synthesis,” in *International Conference on Learning Representations*, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, 2019.

[45] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks,” in *IEEE International Conference on Computer Vision*, Venice, Italy, pp. 2242–2251, 2017.

[46] A. Vakanski, H. P. Jun, D. Paul, and R. Baker, “A data set of human body movements for physical rehabilitation exercises,” *Data*, vol. 3, no. 2, pp. 1–15, March 2018.

[47] Vicon Plug-in Gait Reference Guide, Vicon Motion Systems, 2016.

[48] Y. Liao, A. Vakanski, and M. Xian, "A deep learning framework for assessing physical rehabilitation exercises," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 2, pp. 468–477, Feb. 2020.