Machine learning approaches for 3D motion synthesis and musculoskeletal dynamics estimation: A Survey

Iliana Loi, Evangelia I. Zacharaki, and Konstantinos Moustakas, *Senior Member, IEEE,*

Abstract—The inference of 3D motion and dynamics of the human musculoskeletal system has traditionally been solved using physics-based methods that exploit physical parameters to provide realistic simulations. Yet, such methods suffer from computational complexity and reduced stability, hindering their use in computer graphics applications that require real-time performance. With the recent explosion of data capture (mocap, video) machine learning (ML) has started to become popular as it is able to create surrogate models harnessing the huge amount of data stemming from various sources, minimizing computational time (instead of resource usage), and most importantly, approximate real-time solutions. The main purpose of this paper is to provide a review and classification of the most recent works regarding motion prediction, motion synthesis as well as musculoskeletal dynamics estimation problems using ML techniques, in order to offer sufficient insight into the state-of-the-art and draw new research directions. While the study of motion may appear distinct to musculoskeletal dynamics, these application domains provide jointly the link for more natural computer graphics character animation, since ML-based musculoskeletal dynamics estimation enables modeling of more long-term, temporally evolving, ergonomic effects, while offering automated and fast solutions. Overall, our review offers an in-depth presentation and classification of ML applications in human motion analysis, unlike previous survey articles focusing on specific aspects of motion prediction.

Index Terms—Computer Graphics, Animation, Motion Prediction, Motion Synthesis, Machine Learning, Biomechanical Simulation.

1 Introduction

Motion prediction and synthesis methods are narrowing the gap between handcrafted and automatic animation. To date, the prediction of 3D human motion and dynamics is performed mainly using physics-based methods that produce accurate and realistic solutions, while taking into consideration physical parameters such as joint velocities, forces, torques, etc. For example, in the field of biomedical engineering, human kinematics and musculoskeletal dynamics estimation is performed with great precision by open-source software like OpenSim [1], [2], or computational models like Finite Element Models (FEMs) that simulate the behavior of an individual's lower limbs. Such methods use kinematics and dynamics equations to calculate human movement, joint forces, or muscle function biomechanics given raw human motion capture data (e.g. joint angles and torques) and/or forces. There are a plethora of publications concerning the use of physics-based techniques mainly focused on FEMs, such as [3], [4], that exploit and validate subject-specific FEMs in order to estimate muscle-tendon forces, joint contact forces, and joint contact mechanics in patients with knee prostheses.

However, in the last decades both biomedical and computer graphic engineers have turned towards data-driven (machine learning) approaches in order to obtain faster computation times and more automated, as well as real-time, solutions while estimating human biomechanics. In this paper, we aim to provide a review and categorization of the most recent studies investigating kinematics and dynamics estimation problems, as well as motion synthesis techniques using machine/deep learning, in order to offer sufficient insight into the state-of-the-art and help new researchers in these application domains. Thus, unlike previous survey articles which focus on dedicated problems (e.g., only vision-based motion trajectory prediction using deep learning (DL) [5] or kinematics prediction using predictive modeling and data mining to study the human kinematics of patients with neuromuscular and musculoskeletal diseases [6]), our review summarizes the most recent innovations in three different, yet related, application domains, namely motion prediction, motion synthesis, and musculoskeletal dynamics estimation, offering a more general view to ML-driven human motion estimation.

Most works in motion prediction and synthesis, as will be presented below, implement various machine learning models, which are using human motion capture and movement history data (e.g. previous frame/s or motion state/s such as joint angles) as input to predict or synthesize the pose and/or the joint trajectories of a virtual character. Both motion prediction and synthesis publications are classified into deterministic methods that predict/synthesize a motion sequence, which converges to the ground truth, and probabilistic techniques, which predict/construct all plausible pose sequences of a 3D character based on historical poses and/or control inputs. Motion synthesis works also include physics-based and diversified motion synthesis techniques. Special attention is given to ML-based solutions for musculoskeletal dynamics estimation, which open the path for ergonomically-adjusted motion estimation (e.g. fa-

• *I. Loi, E. I. Zacharaki and K. Moustakas are with the Department of Electrical and Computer Engineering, University of Patras, Greece. Corresponding author: I. Loi. E-mail: loi@ceid.upatras.gr*

Manuscript received month day, date; revised month day, date

tigue modeling), and thus, contribute to the simulation of more realistic computer graphics' character animation. That is because such biomechanics works provide insight into the internal processes of the human body instead of superficially estimating or reproducing human motion without taking into consideration significant movement parameters such as joint and muscle forces. Therefore, a connection between kinematics and dynamics estimation techniques is starting to be explored by state-of-the-art approaches such as Physics-Informed Neural Networks (PINNs) to estimate both internal forces and joint kinematics [7], [8].

This work is organized as follows. In section 2 we provide high-level definitions in respect to motion prediction and synthesis. Moreover, some basic notations are defined in order to provide a broader context of the concepts presented in the related literature. Sections 3-5 present published approaches and applications in motion prediction and synthesis, along with musculoskeletal dynamics estimation solutions. Section 6 includes a comparative analysis between the reviewed works in the aforementioned three topics and some discussion on limitations and potential open issues, while section 7 provides a conclusion to this review. Finally, an Appendix (Appendix A) that outlines our literature search methodology, an Appendix (Appendix B) containing a thorough description of the most popular human motion datasets for motion prediction and synthesis, and an Appendix (Appendix C) with key terms and details on established classifiers, are provided as supplementary material.

2 Background and Rationale

2.1 Goal of Motion Prediction

The main goal of a motion prediction algorithm is given a sequence of input (usually motion capture) data $X = \{\mathbf{x}_1, ..., \mathbf{x}_T\}$, where $\mathbf{x}_t \in R^3$ is the 3D pose element of a human at time frame $t = 1, ..., T$, to estimate the motion (pose) $\hat{\mathbf{y}}_{t'} \in R^3$ at a future time point $t' > T$. A similar formulation of motion prediction or motion forecasting can be found in the highly cited paper [9]. Each 3D pose at time point t is usually visualized on a 3D skeletal structure of the human joints and is parametrized through joint positions (or joint angles).

The motion prediction problem has been studied in many applications concerning kinematics prediction [6] and kinematics prediction combined with motion synthesis (e.g. [10], [11], [12]). Depending on the application, the input of motion prediction algorithms may differ. As stated in [13], many methods are using only a single source of information such as the pose of a virtual character at the current frame [14], [15], but there are also approaches that encode high-level contextual information such as scene interactions (e.g. [10]) to potentially offer more robust predictions, e.g. in combination with motion synthesis (section 2.2). Most of these works exploit recursive deep neural networks [10], [15], [16] which are using joint positions, trajectories, velocities, rotations, a character's pose, etc. of the previous frame/s, as input to the network, in order to predict the future human model's pose or a sequence of motion-derived parameters.

2.2 Goal of Motion Synthesis

The aim of human motion synthesis is to find a model f that can simulate new human movements (solutions) which represent either task-dependent motions obtained from example data (deterministic motion modeling) or stem from a set of randomly sampled latent variables (probabilistic motion modeling). To build the function f, a reference motion set is utilized, $\mathbf{X} = \left\{X^{(1)}, ..., X^{(n)}\right\}$, where n is the number of observations used to capture the variability of the motion pattern, selected usually under certain conditions or activities (e.g. sit on a chair). As the modeling of distinct actions (motion classes) in obstacle-free environments finds limited applicability, especially in computer graphics, motion synthesis methods usually incorporate a combination of input parameters, reflecting not only the digital character's pose and trajectories, but also environment parameters, interactions, and goal-setting parameters. All the different parameters are used in order to generate a novel action pose, $\hat{\mathbf{Y}} = f(\mathbf{X})$.

Both physics-based and data-driven methods for motion synthesis are described thoroughly in previous review studies, such as in [17], where a general overview of 3D human movement generation methods is provided. In contrast to physics-based approaches, data-driven methods produce more realistic and expressive movements, since they use real (prerecorded) movements for training. Nevertheless, they are prone to motion artifacts, such as lack of balance, and still rely on samples "seen" in the training dataset, which renders them incapable of modeling a wide variety of movements [18]. On the contrary, advanced ML techniques and generative modeling approaches are able to synthesize new motion patterns by taking into consideration environmental and time parameters and create movements that are not explicitly defined in their training dataset [18].

3 Motion Prediction

3.1 Short-term and Long-term Motion Prediction

Most works addressing motion prediction focus on short-term [15] and long-term [14], [16], [19] motion sequence forecasting, i.e. the generation of future body poses (represented through 3D skeletons) based on observed past human motion frames with respect to the current frame. The goal of short-term prediction is the estimation of the next-frame pose, whereas long-term prediction deals with predicting poses over a longer horizon or over the next actions. That is, given an input (ground truth during training) pose sequence with spanning time $t = 1$ to τ: $X = \{\mathbf{x}_1, ..., \mathbf{x}_\tau\}$, a short-term model predicts the pose of the next frame $t = \tau + 1$, namely $\hat{\mathbf{y}}_\tau = \mathbf{x}_{\tau+1}$, while long-term motion prediction approaches predict the future poses over the horizon $t = \tau + 1$ to $\tau + T$, thus, the output will be $Y = \{\hat{\mathbf{y}}_{\tau+1}, ..., \hat{\mathbf{y}}_{\tau+H}\}$ [19]. An example of 3D pose skeleton sequences' short and long-term prediction is given in Fig. 1.

3.1.1 Recurrent-based structures

One of the first works on short-term motion prediction that has set the foundation for long-term prediction is presented in [15]. Instead of directly predicting the body pose, joint velocities are estimated by implementing a simple Recurrent

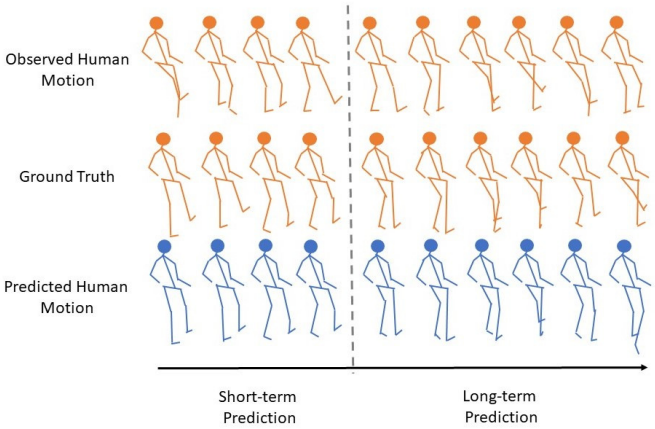

Fig. 1. Short-term and long-term human motion prediction on gait as reproduced from [14]. During inference, given an observed (input) 3D pose sequence (first row) $X = \{\mathbf{x}_t\}$, where $\mathbf{x}_t \in R^3$ for $t = 1, ..., T$, the ML-based framework proposed by [14] produces future pose sequences (red skeletons) at a future time point $t' > T$, which resemble the ground truth (second row). Specifically, the first 4 frames (skeleton poses) correspond to a short-term prediction (predicting a motion sequence with a length of 400 ms), while predicting 10 motion frames constitutes a long-term prediction (predicting a motion with a length of 1000 ms).

Neural Network (RNN) architecture using Gating Recurrent Units (GRUs) as decoder. GRUs do not own an output layer, resulting in training fewer parameters, thus, they can be computationally faster compared with other RNN models like LSTMs [20], [21]. This work focuses on short-term motion prediction but also builds the foundation around modeling long-term motion dependencies. Another short-term prediction approach is presented in [22], where an RNN-based motion prediction framework for short-term (in the span of 1 second time period) forecasting during fencing is proposed. This architecture is specifically designed for mutually interacting characters, such as pair of athletes in fencing videos, and consists of two RNN encoder-decoders, each of them predicting the motion of one of the two characters. This model consists of two RNN encoder-decoder architectures, each of them predicting the motion of one of the two players, since each player's movement varies from the other's, and the models are mutually connected to predict the movement of the players depending on the interaction between them. More specifically, the players' interaction is simulated by feeding the output of the RNN's middle layer of each player to the one of the opposing player, since this layer contains both past and current player's pose information, which is a good representation of the context of motion of the opponent player. This "mutual connections" technique was tested using single-person architectures such as in [9] and [15]. The evaluations in [22] suggest that their framework accurately predicts moves in reaction to the opponent, like avoiding.

Instead of using GRUs for long-term prediction, the authors of [16] proposed a method that explicitly encodes human and animal anatomy by modeling the skeletons as a kinematic tree consisting of one or multiple kinematic sequences and based on the mathematical formalism of Lie algebra [16]. An alternative to a GRU-based RNN [15] is proposed in [14], where the authors used a Modified Highway Unit (MHU) as a component of the recurrent prediction layer (decoder). In this work [14], MHUs were used to differentiate at each pose the joints participating in a movement from the ones that do not participate, and thus are claiming to be more efficient for long-term motion prediction. A recent work [19], which shows improved results over the previous short-term prediction approach [15], uses GRUs for the description of long-term dependencies in motion. The network is trained using a continual learning training scheme, where a robust general representation of motion is first learned based on training samples from simpler diverse tasks, and then only the decoder module is fine-tuned in order to predict the motion of a specific (new) human subject, mitigating the problem of training from scratch.

3.1.2 Attention- and Graph-based Models

A different approach to both short-term and long-term motion prediction which is advertised to outerperform [14], [15], tries to discover repeated motions along a human's movement by applying an attention-based feed-forward network that predicts future motion sequences by comparing current sequences with similar (to the current motion) past pose sub-sequences [23]. The method in [23] improves over previous work [14] in that it does not calculate similarities between static poses (that are not unique within each action) but captures the spatiotemporal dependencies in the data using motion attention. Specifically, it estimates future motion through weighted aggregation of historical motion sequences, combined with the current pose sequence, and introduced as input to a Graph Convolutional Network (GCN). According to the authors, this approach leads to better learning. Inspired by [23], the authors in [24], used a similar technique to produce motion prediction mappings of two people who interact with each other. In [24], a framework containing two pipelines was constructed each of them consisting of an attention model for learning temporal relationships and a GCN predictor for extracting spatial dependencies between the joints of the pose skeletons, likewise to the technique used in [23]. Moreover, for modeling the interaction between two persons a Cross-Interaction Attention (XIA) model was used to generate the future pose of each person based on the movement of the other. Specifically, XIA uses multi-head self-attention to generate weights shared between the two predictors that model the movement of each person. For the purposes of this work, the authors recorded a dataset that contains 115 sequences of highly interacted Lindy-hop dancing poses, called the ExPI (Extreme Pose Interaction) [24].

A work that can be assumed as a continuation of [23], is [25] where a graph generative model consisting of GCN layers was developed, which performs both short- and long-term prediction by taking into consideration the natural connectivities of the human joint pairs of the skeleton pose (treating the latter as a dynamic graph). For this purpose, two parameterized graphs are constructed; one to explicitly learn (as weights) the natural kinematic connections between joints of 3D skeletal sequences, and another one to implicitly learn the relationships between joints that are not geometrically connected. This framework dynamically learns the relationships between the joints in 3D skeleton

sequences, and, thus, presents a more flexible and accurate approach compared to approaches, such as [23] that consider the 3D poses static. A work that also uses graphs to model the relationships in the 3D skeletal poses is [26]. This framework is an encoder-decoder architecture called Dynamic Multiscale Graph Neural Network (DMGNN), based on multiscale graphs that have nodes that represent body parts at various scales and the edges of the graph represent the relationships between these parts. In particular, the encoder contains a sequence of Multiscale Graph Convolutional Units (MGCU) each one of which corresponds to a multiscale graph, to extract temporal and spatial features at multiple scales from the input motion sequences. Also, the decoder includes Graph-based Gated Recurrent Units (G-GRUs) to generate predictions. Similarly to [25], in [26] the graphs are learnable and dynamic since their topology is changing across the model's layers.

More recent works that utilize GCN-based frameworks are [27] and [28]. In [27] a Gating-Adjacency GCN (GAGCN) is used as an encoder to learn both joint dependencies and temporal relationships from past pose sequences. Each GAGCN encoder layer uses both a spatial gating network and a temporal gating network, to produce blending coefficients which are used to blend the trainable spatial and temporal adjacency matrices in order to learn correlations between joints and time frames. More specifically, gating networks classify motions by automatically computing the probability of the action class that the observed pose sequence belongs to [27], and, thus, can be used to enhance the generalization of neural networks in more than one movement. Blending coefficients and gating networks will be re-explored in Section 4. Supplementary, the GAGCN scales the number of adjacency matrices to balance spatiotemporal features and further fuses these features to explore the cross-correlations that reside between them in past motions. The method proposed in [27] outperforms previous long-term prediction works such as [23] and short-term ones, e.g. [15]. As for [28] the authors use "initial guesses" of the future pose sequences to increase the prediction accuracy progressively. For this purpose, the authors employ a two-stage prediction model, which consists of two motion prediction networks: the init-prediction network that predicts the mean of future poses as an initial guess, and the formal prediction network that is fed with a concatenation of the input (past poses) and the predicted (guessed) pose sequence to generate the final result. This framework is used as a base to create a multi-stage network, where each stage generates a more smoothed version of the ground truth motion sequence. These smoothed motions are used as intermediate targets to progressively produce better and better guesses, which will result in a more accurate final future pose sequence. What is more, each stage uses spatial dense GCNs (S-DGCN) and temporal dense GCNs (T-DGCN) to extract spatial dependencies in 3D poses and temporal correlations in joint trajectories, respectively, whereas in [25] GCNs are used only to process spatial features.

3.2 Probabilistic Motion Prediction

The works reviewed so far that address the motion prediction problem, are deterministic approaches, meaning that given an input human motion, they predict a deterministic future pose sequence that regresses towards the mean pose. On the contrary, there are probabilistic (or stochastic) methods, which predict diverse possible future motions based on a single historical motion sequence [29]. Such works usually use Variational Autoencoders (VAEs) [30], [31], which consist of an encoder-decoder architecture. The encoder takes as input data $\mathbf{x}$ and transforms these data into a latent representation $\mathbf{z}$, meaning that it approximates the posterior distribution $q_\phi(\mathbf{z}|\mathbf{x})$, where ϕ are the parameters of the encoder. The $\mathbf{z}$ expresses the variation in data and is given as input to the decoder that produces a reconstruction of the initial data, $\hat{\mathbf{x}}$, by learning $p_\theta(\mathbf{x}|\mathbf{z})$, where θ are the parameters of the decoder [30], [32]. The loss function of the model consists of a reconstruction term that aims at increasing the performance of the model and a regularization term. The latter regularizes the latent space, meaning that it brings the distribution of $\mathbf{z}$ closer to a standard normal distribution, in order to both prevent overfitting and give the ability to the model to produce new data [30]. Contrariwise, the simple autoencoders can have an irregular latent space (e.g. decoded data from close latent space points can be quite different, etc.), thus, a generative process cannot be constructed upon them. An extension to VAEs, are the Conditional Variational Autoencoders (CVAEs) that use additional information, such as a conditioning variable or a specific past pose sequence, to produce $\hat{\mathbf{x}}$. Thus, conditioned on a conditioning variable c, the encoder of CVAE becomes $q_\phi(\mathbf{z}|\mathbf{x}, c)$ and the corresponding decoder $p_\theta(\mathbf{x}|\mathbf{z}, c)$ [32].

In contrast to the works mentioned in Section 3.1 that use 3D skeletal representations to model pose sequences, in [29] a time sequence of 3D markers (3D joint locations forming a sparse point cloud) is used to fit into a 3D human body mesh to produce realistic motion predictions. Particularly, in this work, a CVAE, called MOJO, is developed to both perform motion prediction and also produce more realistic results by retaining the full temporal resolution of the input sequence and also integrating high-frequency motion components into the predicted motion. In [32], [33] not only future motion is generated stochastically, but diversity and context of motion, respectively, are imposed via stochastic processes. Specifically, in [33] a more stochastic way to enforce diverse motion is discussed: instead of combining historical pose sequences (or model's hidden state) with noise vector to model stochasticity (and further produce plausible future motion) in rather a deterministic way, the authors proposed to randomly select and shuffle a part of the hidden state of the model with a random vector in every training iteration. As a result, the model is forced to take into consideration this random noise in order to enhance the diversity in future pose sequences. This technique is integrated into a recurrent encoder-decoder network with a CVAE block, where the RNN encoder uses past pose sequences to produce the hidden state, part of which is combined with noise as aforementioned to create a "perturbation" vector. The latter is fed into a VAE decoder whose output is given to an RNN decoder to produce the future motion [33]. As a continuation of their previous work, [33], in [32] the same authors introduce a variational model that aims both at diversity in motion forecasting and taking into account the context of the future 3D poses, so instead of conditioning the encoder, as in [33], they also condition the decoder. To do so, the authors of [32]

developed a CVAE consisting of two autoencoders, i.e. one VAE fed with past 3D poses (conditioning poses) that learns the distribution of the conditioning signal and another VAE that learns the distribution of the latent representation of the future motion depending on the condition. By sampling the latent representation of the input based on historical 3D poses, the latent variable is imposed to have relevant (to the conditioning signal) information, and, thus, produce contextually plausible future motion.

A state-of-the-art probabilistic motion forecasting framework is Motron [34], which models the multi-modality of humans. Motron follows an encoder-decoder architecture as in the above similar works, whereas in this model latent representation samples $\mathbf{z}$ are combined with the hidden state of the encoder before being fed to the decoder, in order to induce diversity in motion. To model motion multi-modality, the distribution over the latent variable $q_\phi(\mathbf{z}|\mathbf{x})$ is used. In contrast to CVAEs, the latent variables produced in this work do not explicitly learn a representation for the decoder, but the decoder implicitly learns to distinguish future output poses [34]. What is more interesting about this method, is that it was tested under handling data with occlusion and produced confidence values to measure its uncertainty of predictions. Another recent publication addressing motion multi-modality is [35]. In [35] a multi-objective CVAE-based model to balance both accuracy and diversity in motion forecasting was implemented. To this end, this network infers two different prior distributions; the accuracy $q_{acc}(\mathbf{z}|\mathbf{x})$ and the diversity one $q_{div}(\mathbf{z}|\mathbf{x})$. By sampling from the accuracy prior distribution, the distribution of the input dataset is estimated, and different pose sequences are explored by learning the diverse prior distribution. Once again this framework follows an encoder-decoder architecture; the encoder extracts temporal features of motion trajectories using RNN and feed-forward networks, and the decoder predicts the future poses. To further enhance the motion diversity, a short-term oracle CVAE-based model infuses pseudo-ground-truth multi-modality into the proposed model [35]. Lastly, a recent work that addresses the problem of action-drive probabilistic motion prediction is [36]. A CVAE-based model was created in [36], which, conditioned on historical motion sequences and action labels, predicts possible plausible motions, unlike works that focus on conditioning motion prediction on a single action. Two different temporal encoding structures, an RNN-based and a Transformer-one were used to build the VAE. The main contribution of this work is the prediction of multi-action motion sequences (i.e. sequences that contain multiple actions per sequence, such as drinking and passing the bottle) while providing smooth transitions between diverse actions in the motion and allowing the predicted motions to have varying lengths. To simulate action transitions in the training dataset, the authors in [36], proposed a training method that combines the creation of synthetic motions, from past and future motion sequences from different actions, with the use of a weakly-supervised technique to aid in creating smooth transitions between action classes. This framework also incorporates a variance-based method to generate pose sequences of different lengths.

3.3 Controlled Motion Prediction

A subcategory of motion synthesis and motion prediction studies targets the *controlled motion generation*. These methods generate a motion conditioned on a control signal that, for example, predefines the direction, style, or velocity of the movement, etc. and thereby achieves interactive control over the generated motion [37]. Specifically, controlled motion prediction produces future motion poses (at next time frame $t+1$) based on past observed poses and control signals in order to control the movement of different body parts [38] or predict transitions between different actions [39] or frames [40]. On the contrary, controlled motion synthesis focuses on synthesizing new movements (at current time frame t) in a controlled fashion (conditioned on specific trajectories, velocities, goals, etc.), rather than free motion generation methods (see Section 4.1). A work that is a representative example of controlled motion prediction is presented by Harvey et al. [40]. In this paper, adversarial recurrent neural networks (RNNs) were used for the automatic generation of the transition motion between an initial and a target pose. In particular, the authors developed a motion prediction RNN, to which they applied two different embedding modifiers at each timestep of inference: (i) a *time-to-arrival embedding* was applied on the hidden representation of all inputs in order to facilitate the handling of in-between pose transitions of different lengths for a single model, and (ii) a *scheduled target noise* vector was introduced aiding the model to withstand distortions in target poses. The time-to-arrival embedding is evolving from the target pose backward to the initial one, in order to permit the recurrent layer of the network to have continuous insight into the number of timesteps needed to reach the target pose, thereby helping to produce a more dense and smooth motion. The scheduled target noise vector feeds the RNN with distorted target poses when a long transition is about to start, in order to aid the motion generator model to reach the correct pose. To increase the quality of the synthesized motion, a generative adversarial network (GAN) was added [40]. Similar works to the one in [40], which however focus more on motion synthesis than prediction (like [37]), are presented in Section 4.1.

Other works include [38], [39], where in [38] a unified deep generative model was developed for generating both controllable and diverse motion sequences. This model sequentially predicts the motion of different body parts giving the ability to retain the movement of some parts of the body by fixing the latent representations of these parts, while predicting diverse solutions for the other parts of the body (by varying their latent representations), and, thus, offering controllability. So, this model learns a pose prior, instead of a motion distribution that most aforementioned probabilistic methods do, which facilitates the learning of diverse motions. Also, a joint angle loss is used to penalizethe predicted motions according to restrictions of the anatomical structure of the human body, meaning that joint angle values are limited to a certain range, which range is defined by processing 3D human poses. The pose prior along with the joint angle loss provide temporal smoothness and more realistic motion [38]. Moreover, in [39] an Aggregated Multi-GAN model was presented, that gives the ability to control

motion prediction across actions and further customize the forecasted movement. To do so, the authors of this work developed local GANs to guide the motion of different body parts and a global GAN to aggregate the complete motion to maintain synchronicity and balance between the limbs (kinematic chains) of the animated character.

4 Motion Synthesis

4.1 Controlled Motion Synthesis

4.1.1 Deterministic Approaches

4.1.1.1 Recurrent Architectures: One of the most prominent papers in this scientific area [10] proposes a deep auto-regressive algorithm, called the Neural State Machine (NSM), which predicts in real-time the character's movement and interaction with scene objects given a specific goal (e.g. sit on a chair in the opposite side of the scene). More specifically, the NSM enables automatic transition between different high-level action states (e.g. run to sit) in order to reach a user-specific goal state. NSM consists of two neural networks: the gating network and the motion prediction network. The gating network computes a set of parameters, called blending coefficients, which are mixed with the network's weights to predict the transition from one action to the other based on the given goal and the phase of the motion signal (phase defines the different stages of motion). The motion prediction network is responsible for synthesizing the character's pose and movement in the current and future frames based on previous frames' character and scene information. The NSM as described in [10] is presented in Fig. 2. A work similar to NSM is [41], where a fully automated approach based on adversarial imitation learning generates a control policy that permits the character to achieve a specific goal in a virtual environment while imitating motion styles from the training dataset. In particular, the proposed model incorporates an adversarial system, called adversarial motion prior (AMP), which can specify the low-level motion style (e.g. jumping, running) of the character while performing a task. A dataset of unstructured motion clips is exploited by an RL-based method to automatically select the high-level tasks that the character should perform. The main innovation of this model is that it bypasses the need to manually design or tune reward functions for different motions [41].

The same authors of NSM published two more works as a continuation of the work in [10] - one in 2020 [11] and one in 2021 [12]. In [11] a novel framework is proposed that dynamically synthesizes character-character, character-object, and character-scene interactions containing multiple contacts. The main architecture of their framework resembles the one of NSM but is enriched with a local motion phase feature and a generative control scheme. In more detail, in contrast to the NSM where action movements are synchronized via a single global phase variable, in the latter work local motion phases of each body (skeletal) segment being in contact with an object (such as a ball) are used to train the proposed model. This local phase feature helps the neural network to learn the asynchronous motion of each body segment and its interaction with the environment and scene objects. Moreover, in order for the neural network to produce fast and complex movements, like the ones

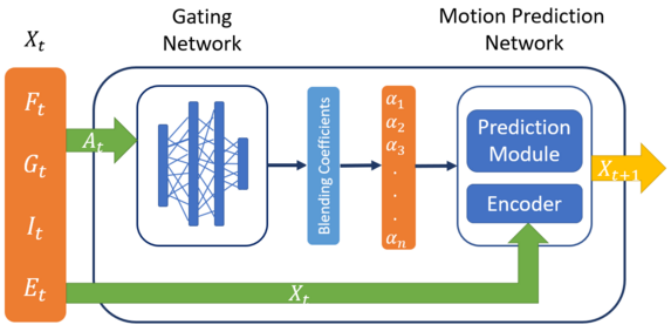

Fig. 2. The architecture of NSM as presented in [10]. F is the frame input consisting of the character's pose and past/future trajectory info, and G is the goal input composed of goal positions and orientations as well as the action that the character has to perform when reaching the goal. Furthermore, I and G are geometry scene information, and $\{a_1, ..., a_n\}$ are the weights of the Motion Prediction network, which are mixed with the blending coefficients that the Gating Network computes. Moreover, $A_t \subseteq X_t$ is a subset of the current input, which is fed to the gating network to produce the blending coefficients. As for the motion prediction network, it is fed with the character's pose, trajectory, and goal data from the current frame t, and predicts those parameters for the next frame $t + 1$, thus, producing the virtual character's predicted pose, X_{t+1}. For more information concerning the Motion Prediction and Gating Networks comprising NSM, please refer to Sections 3.1 and 3.2 of [10].

used in basketball, the authors present a generative control model. This model is an encoder-decoder network that uses input control signals from a gamepad and generates a more precise sequence of control signals, in order to produce variations in movements and higher-quality motion.

In [12], another work of the same authors, a deep learning framework for character animation layering is described. This system can synthesize brand-new combinations of martial art movements and close-character interactions from a variety of motion skills given a reference motion and user input control signals. The architecture of this novel framework builds upon their previous motion synthesis work [11] consisting of a gating and a pose prediction network used to reproduce movements from motion capture data and generate new movements. Following the training of the model, various control modules are created from subsets of data. These modules can have various forms (neural networks, physics-based simulations, animation clips, etc.) and are used to produce future movement trajectories of the joints participating in motion and, thus, aim to drive motion synthesis. Then, these trajectories are given as input to a control scheme in order to be layered by additive, override, or blending operations. The output of this control interface is fed to the motion synthesis network to finally produce a novel full-pose for the character from the current frame to the next one. This model can be used for both online and offline motion synthesis like their two previous works. In the most recent work [42] of the team of [10], [11], [12], a new model called the Periodic Autoencoder was created. The main goal of this neural network is to learn spatiotemporal periodic embeddings from a huge amount of unstructured data without supervision. In particular, this network transforms the motion space into a learned multi-dimensional phase feature space by using a temporal convolutional autoencoder. This model treats the 3D character's

motion as different local phase signals, corresponding to the movement of the different body parts. The model enforces these signals to be in a periodic form and exploits this local periodicity in order to extract the phase feature space. The model's capability to be used along with similar to previous frameworks (the local motion phases neural network introduced in [11] and the one in [43]) for a variety of character animation tasks such as predicting or synthesizing locomotion patterns, to highly-complex music-driven dance moves and character-object interactions as well as improve the quality of motion synthesis, was also explored in [42].

Furthermore, it is worth mentioning that the predecessor of NSM [10] is a real-time motion synthesis regression framework called the Phase-Functioned Neural Network (PFNN) [44]. Similarly to [10], this network uses the previous pose of the character, a single phase parameter, the user control signals, and scene geometry as input. The network's weights are modified according to a simple phase function, giving the ability to produce various movements for the character in order to achieve the user-desired task. Moreover, this framework is trained using a vast amount of data including different movements (e.g. walking, climbing, running, etc.), which are fitted to heightmap data (e.g. terrain) from virtual environments in order to adapt the character's motion to a variety of geometric environments.

More recent deterministic controlled motion synthesis approaches include [45], [46], where the synthesis of complex 3D dance moves synchronized with music pieces is realized. In [46], a full-attention cross-modal transformer-based model was developed, which is fed with a music clip, an initial part of the motion (called seed motion), and some specific (key) pose moves that are expected to belong in the dance sequence. For this purpose, two single-modal transformer encoders were created, one to produce embeddings of music clips and one to encode 3D pose sequences with full attention across all time steps, respectively, as well as a cross-modal transformer decoder to learn the correlations between the cross-modal data and fuse their representations in order to synthesize dance sequences based on key poses. Aside from key pose mappings, the cross-modal transformer models local position embeddings, which are representations of the relative positions of the key poses in the generated motion. The latter create a position prior that enforces the synthesized poses at the corresponding (key) frames to be consistent with the sample key poses. Both key pose and local position embeddings control the dance movements that are generated by the model. Another music-driven motion synthesis work is [45], which focuses on imprinting the global context of a dance genre, instead of exploiting local features of the dance sequence such as in [46]. The framework created for the purposes of [45], is a hierarchical three-level model. The first level is an auto-conditional LSTM model, meaning that is conditioned both on the music and the movement patterns to produce temporally consistent motions that are synchronized with the music's beat. The second level aids in clustering consecutive pose sequences to certain distributions, thus, controlling the dance moves and introducing more naturalness to the produced motion. This level also enables motion diversity by applying scaling and translation transformations to the joint rotations of the input pose sequences. Finally, the last level fixes the order of the movements and further enforces the generated motion to follow the distribution (global structure) that characterizes a specific dance genre, i.e. choosing the pattern in the dance [45]. The second and third levels use traditional deep learning methods such as Multi-Layer Perceptrons (MLP).

4.1.1.2 GAN-based Methods: Even though generative adversarial networks (GANs) are a probabilistic modeling approach [47], they can be utilized as part of deterministic models to create a diverse training set upon which the output of the model will be conditioned. Such examples that incorporate generative adversarial training are [48], [49]. Just as in [10], [50], a generative network that models the movement of a virtual human in a 3D scene conditioned on the character's previous pose and the environment, was developed in [48]. This framework consists of three parts, where the first is a scene encoder, which is fed with a scene in an RGB image format to produce the scene's geometry-aware feature (visual semantics and structure), and the other two are a generator and a discriminator constituting a GAN-based learning approach. The scene embedding that the first component of the model generates is used to condition the model's results. Specifically, the generator learns the latent distribution of the motion trajectories that are used to sample trajectories based on which the pose distribution is produced. By aligning diverse poses and sampled trajectories, a novel 3D pose sequence based on the virtual scene is synthesized. As for the discriminator component, it consists of four discriminator modules; the trajectory and pose discriminator aid in creating smooth and continuous pose sequences, while the projection and context discriminators impose the generated motion to follow the global structure of the virtual scene (e.g. the floor) and the local structure (e.g. objects to be avoided) at each frame, respectively. In contrast to the aforementioned works that require large and well-structured datasets to be trained, in [49] a generative model that is able to synthesize new and diverse pose sequences trained only on a single short motion series was constructed. This model is called GANimator and consists of GAN-based components, where the encoder of each GAN part incorporates skeleton-aware convolution layers [51] (see section 4.1.2 for more details on skeleton-aware architectures). Each component of this model is fed with the output of the previous part and a random noise vector to perform upsampling upon the input motion sequence, which results in progressively increasing the temporal resolution of this single training motion. Also, these layers are automatically adjusted to preserve and follow the skeleton structure of the animated subject, thus, giving the flexibility to synthesize motion for both humans and animals with any number of limbs. What is interesting about this work is that it produces both controllable motion sequences and uncontrollable ones that take after the core frames of the original input sequence. The unconditional motion series are synthesized based on random noise and usually simulate crowd motion and motion mixing and editing, whereas controllable movement can be generated conditioned on user inputs. Furthermore, this model can be used for both motion editing and interpolation, and style transfer.

4.1.2 Probabilistic Approaches

Besides the deterministic motion synthesis approaches, there are studies using probabilistic methods for controlled motion generation [37], [52]. While deterministic models lead to repetitive characters in applications, the probabilistic models generate different motions upon each subsequent invocation, even for the same control signal, and also provide a measure of the likelihood for each possible motion. Combining probabilistic methods with controlled motion generation approaches gives the ability to interactively control the synthesized output motion based on a specific condition (control input signal e.g. current pose, the direction of walking, etc.).

4.1.2.1 GAN Models: In [53] a system utilizing GANs was developed to synthesize realistic character reactions to another virtual character's or user's motion. The model is an attentive recurrent network (an LSTM), which allocates its weights to the preferred content from the input information, such that the main character can focus (pay attention) to the movement of another character and accordingly react. The generator has an encoder-decoder architecture, with the encoder being a part-aware LSTM that learns the encoding of the observed motion by separately modeling the body part-level dynamics of the input character (the one performing the action to which the network must react). The decoder is an attentive LSTM that, based on the encoder information, temporally aligns the decoded reactive motion with the input character. Furthermore, the discriminator consists of LSTMs and has two tasks: to differentiate the generated reactions from the natural ones and to identify the class label of the interaction. The recognized classes can be then used to train the generator in a supervised manner. Similarly, Mourot et al. [54] present a method that estimates complex movements, such as jumps of a 2D avatar, by incorporating in the architecture a GAN and an encoder that help to learn mappings from human pose sequences to GAN's latent space. This method has the ability to upsample the number of joints in each pose sequence in order to correct any missing or occluded joints.

4.1.2.2 Variational Autoencoders: An autoregressive conditional variational autoencoder (VAE) was used in [52], producing a distribution of future poses based on a set of stochastic latent variables. The VAE is controlled via a reinforcement learning (RL) model to produce the desired motion. More specifically, the stochastic latent variables of the VAE define the action space of the movement. The RL model learns control policies that use this action space to govern the VAE in accordance with a reward function that defines the tasks/goals of the character in motion [52]. What is more, the VAE consists of an encoder and a decoder which work in concordance to generate natural pose transitions; the encoder produces a latent representation of high dimensional motion transition information, while the decoder generates the next pose based on this latent representation as well as a condition pose (i.e. current pose) [52]. In this work, the task is learned separately from the motion dynamics, which enables the learning of various control policies using the same motion model, whereas direct prediction methods predict final task-dependent motions from example data.

Variational Autoencoders were also used in [55], [56], [57] for synthesizing novel motion. A two-level hierarchical motion VAE (HM-VAE) that produces mappings of human motion into global and local latent space representations simultaneously, was introduced in [55]. HM-VAE's functionality focuses on extracting both local features, which capture the motion of each body part, and global latent space embeddings, which model the global dependencies between all joints. The model follows an encoder-decoder skeleton-aware architecture to learn over the human skeletal structure (inspired by [51]), where both the decoder and the encoder consist of skeleton convolution, skeleton pooling and skeleton unpooling layers. Explanatory, the skeleton convolution maintains the plentitude of joints in the skeleton structure, while it downsamples the duration of each pose sequence. Skeleton pooling is applied between pairs of connected bones to merge their features and further reduces the spatial features of the input to produce better motion representations, whereas skeleton unpooling does the reverse procedure. Moreover, to aid the construction of the global latent space, a trajectory prediction module, similar to the above-described structure was added to the model. What is interesting about this work is that this framework is task-generic, meaning that it can be used in various tasks, such as synthesizing motion from incomplete observed 3D pose sequences as well as complete corrupted animations (motion interpolation and completion), or in video human pose estimation. A work similar to [55] is [56] where the CVAE-based model was used in motion synthesis, prediction, interpolation, completion, and spatiotemporal recovery tasks. In particular, this model processes each input 3D pose sequence as a masked motion sequence, where the masked regions of the motion are the to-be-generated frames, which are synthesized given the unmasked regions as conditions. Thus, the CVAE extracts a latent distribution of the missing motion regions, from which diverse motions are sampled and plausible motion sequences can be produced. To do so, the model uses two encoder-decoder pipelines as illustrated in Fig. 3; one fed with the ground truth of the masked regions to produce latent representations that are combined with unmasked frames' features to reconstruct the ground truth motion sequence and another one that uses the input conditions to produce different possible motion sequences. This framework also incorporates a module called Action-Adaptive Modulation (AAM) applied to the normalization layers at the decoder level, which exploits motion semantics (i.e. action labels, which are used to learn the parameters of the normalization layers) to better control the style of the synthesized motion. A cross-attention mechanism employed between the features extracted from the encoder and the decoder is also integrated into this model, to produce more realistic and globally coherent movements based on long-term temporal dependencies. As in transformer networks, multi-head self-attention is used to extract the long-term correlations [56].

A state-of-the-art work is [57], where a model that combines Recurrent Transformers (RT) with VAEs, named RTVAE-multi, was developed for the simultaneous generation of multiple action human motion sequences. For each action, the encoder of the network uses a concatenation of 3D human pose sequences with the action label features

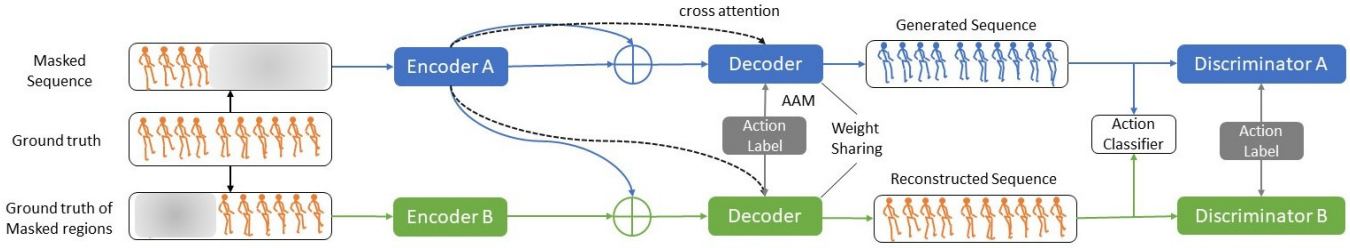

Fig. 3. The architecture of the CVAE-based model proposed in [56]. The model consists of two branches, where the top branch is given the unmasked frames' parts of a motion sequence as input in order to generate plausible motion sequences, and the bottom branch is fed with the ground truth of the masked parts of the same motion to extract features that are blended with the latent representations produced by the upper-branch to reconstruct the complete ground truth pose sequence. As AAM is denoted the Action-Adaptive Modulation mechanism that is utilized to determine the style of generated sequences, while an action classifier is used to spread motion semantics (action labels) through the whole motion sequence. Figure reproduced from [56].

along with the previous hidden state, to predict latent distributions for every action in the sequence. When the encoder processes all the frames of the input pose sequence, then, multiple latent vectors, where each one corresponds to one sub-sequence of an action, are sampled from the latent distributions. The decoder of the architecture uses these vectors stacked with one another, along with sequences representing positional representations of a particular action, to synthesize the input motion. The vectors of all actions are stacked in order to enforce a continuous and temporally coherent representation of the whole input sequence. The decoder can further produce a motion sequence of any length since it does not depend on the timestamp when each action sub-sequence ends. Therefore, the main contribution of the work in [57] is that this model produces realistic human movements with an arbitrary number of both actions and frames.

Another publication in probabilistic motion generation [37], presents a generative and autoregressive controllable motion-data model based on normalizing flows, named MoGlow. MoGlow produces the next pose of the character's movement by drawing a random sample from a simple distribution (e.g. Gaussian), and then it passes it through a neural network in order to transform it in a non-linear way, similarly to a conditional GAN. As a result, the simple initial distribution is transformed into a distribution with high complexity that fits the distribution of the next pose in data [37]. Furthermore, MoGlow is causal, meaning that the output pose is not dependent on future poses of control inputs but only relies on prior poses and relevant control signals. The latter implies the absence of algorithmic latency which is crucial for interactive character animation and control.

A publication that is close to the Neural State Machine [10] is [50]. In this work, a probabilistic model, called the Scene-Aware Motion Prediction method (SAMP), was created in order to predict and synthesize the motion of virtual characters based on a specific goal, while moving and interacting with objects in an enclosed virtual scene. Specifically, this framework consists of three components, the MotionNet, the GoalNet, and a Path Planning Module. The MotionNet is an autoregressive CVAE fed with the previous state (character's pose, trajectory, and goal) and the geometry of the target object (the one with which the virtual character will interact based on the goal). Given the input (historical and current states as well as the object), the encoder of the CVAE extracts a random latent vector at each frame, which is given along with the previous state and the target object, as a condition to the decoder in order to generate the next state. The decoder follows an architecture that resembles the one of NSM [10] since it consists of a gating network that produces blending coefficients and a prediction network. SAMP can be generalized to different objects via using another CVAE, called GoalNet that produces possible contact positions and orientations on the target object. To further give the ability to the virtual character to freely navigate in a scene with multiple objects while avoiding obstacles, a Path Planning Module, i.e. an A* algorithm, is employed to calculate the best path that the avatar has to traverse without hitting on an object in a cluttered scene. It is clear that GoalNet is first employed so as to predict the goal's position and orientation, then the path planning algorithm runs and lastly, MotionNet generates the next character's state.

4.1.2.3 Diffusion Models: A quite recent trend is the use of diffusion models in probabilistic human motion synthesis. One of the most prominent works in this application domain is [58], where a Motion Diffusion Model (MDM) is implemented in a low-resource framework and trained on a single mid-range GPU. MDM is a transformer-based model that produces motion sequences conditioned on either natural language, audio clips [45], [59] or actions [60], while it can also generate motion without relying on conditions. The diffusion process includes noising the input motion sequence and producing the denoised version of the signal as output using a transformer encoder. Unlike traditional diffusion models, at each step, MDM predicts the sample and not the noise, which enables the use of geometric losses (e.g. foot contact loss) in diffusion that mitigate motion artifacts leading to high-quality motion. Similarly to [58], in [61] a computational-cheap model for the prediction of movement rather than noise, was developed. In this work, the Single Motion Diffusion model (SinMDM) is introduced, which is trained based on a single motion input sequence to model movements of any skeletal topology and length. SinMDM was examined under two different neural backbones, where the most notable one is a transformer with local attention layers to temporally min-

imize the receptive field. Learning local motion sequences enables the creation of diverse movements and prevents overfitting. This architecture has a variety of applications including style transfer, modeling long motion sequences, temporal or spatial in-betweening, etc., and demonstrated better performance compared with the work in [49], which constitutes another single-motion technique.

Unlike task-specific works [58], frameworks that offer multitask unification [62], [63] are also present in this category. In [62] a model called MoFusion for unifying a wide variety of motion synthesis tasks such as diverse motion synthesis based on data from multiple sources (audio, text), motion in-betweening, etc., was presented. Like the aforementioned works, MoFusion's main component is also a transformer network that incorporates cross-attention mechanisms to cope with diverse input signals. This framework is pretrained as a diffusion model, which gives the ability to the model to handle partial or whole-body motion synthesis as well as simultaneously generate movements that match multiple sources (e.g. text and music) without the need for training instances that contain multiple sources' labels. Moreover, the authors in [62] showed that pretraining their model reduces overfitting and improves performance. Another interesting work is the one in [63], where a multi-task diffusion model represents motion content and style into a common latent space in order to unify motion generation with style transfer tasks. This multi-task framework predicts different aspects of human motion (e.g. motion trajectories, joint angles, etc.) along with the noise by incorporating more neural networks in order to provide local guidance. However, this model is also optimized by global guidance using a discriminator and physical regulations in order to produce more realistic 3D poses. Lastly, in [64] a physics-guided diffusion model (PhysDiff) was developed based on MDM [58], by integrating a physics-based motion projection module in the diffusion process to reduce motion artifacts (e.g. ground penetration and floating). This module uses a motion imitation policy to guide the movement of a character through a physics simulator at each diffusion step, in order to project the generated (denoised) pose sequence into a physically-plausible motion.

4.2 Physics-based Motion Synthesis

Another wide category of motion synthesis methods is physics-based motion synthesis, which is basically controlled motion generation taking into consideration physical quantities such as joint velocities, forces, torques, etc. The better physical parameters selected for the animated character, the more physically-realistic variations of the reference motion are learned by the motion parameterization policy. In physics-based motion synthesis methods, the action space is well-defined and often comprised of joint torques [52]. For example, in [65] joint torque limits are set, so as to achieve more natural motion parameterization. More specifically, a deep policy network that learns a family of motor skills based on a single motion clip, was developed. This algorithm incorporates a deep network that learns motion parameterization that associates motion parameters to their motions, in order to learn a movement that imitates the input motion. This algorithm also includes a continuous-time reinforcement learning network to model both temporal and spatial variations of the reference motion. The main innovation of [65] is that motion parameterization is learned simultaneously with different motor skills, which improves both computational efficiency and the quality of the resulting motion. In [66] a residual force control (RFC) was integrated into existing RL-based control models in order to overcome the problem of mismatch between real human motion and animated character motion, which does not allow a system to synthesize realistic and complex human movements. The RFC learns during RL training to apply external residual forces to the 3D character in order to better reproduce highly sophisticated motions like a ballet dance.

The works in [67] and [68], which are similar to [65] and [66], also belong in this category. In the latter two works deep RL models learn control policies for physically plausible movement as motion imitation tasks. The work of [67] is worth mentioning, where the motion of a user-controlled physics-based character is interactively produced based on motion capture data, in order to responsively meet the user's specified control changes in real-time. Specifically, a two-step approach to motion synthesis that combines a high degree of responsiveness, while maintaining the natural visual qualities of human motion and balance, is proposed. The first step is a motion-matching network that implements kinematic controllers. The kinematic controllers' utility is to find and select frames from an animation database (unstructured motion capture database in this work), based on certain physical and user control input requirements. The frames that satisfy these requirements are continuously combined to synthesize a virtual character's motion. In this work, the requirements are features like the character's future trajectory positions and orientations, animation style, and continuity of foot placements. The second step is a reinforcement learning network which provides a feedback system to maintain character balance. Based on this, the virtual character moves and balances relying entirely on its own strength, thus resulting in a realistic interaction with the environment. Both the virtual character and the objects in the environment used to test this framework were given physical properties like mass, friction, etc., and other physical parameters were set, such as upper limits for torques and velocities in the joints.

Motion prediction is combined with physics-based motion synthesis in [69] through the development of a motion generation model that combines VAEs and Inverse Kinematics. The VAE uses a new sampling technique in the latent space to compensate for the motion detail loss when the dimension of the latent space is low, and also for sampling difficulties due to sparsity in training data caused by a high dimensional latent space. This allows to produce plausible motions even from a small amount of training data. The IK method is used to enhance the synthesized pose by matching selected points in the body to a desired pose (e.g. as defined by motion capture). This IK method requires less effort since it provides a target sampling space for the pose in every keyframe, instead of manually annotating specific target positions in all frames of the motion, as performed in traditional IK techniques. Furthermore, this model incorporates a motion correction technique based on i) imitation learning with a physics simulation, and ii) a motion debi-

asing module. Imitation learning aims at training a policy that modifies the next pose in a way that resembles the goal pose. Specifically, the imitation learning scheme proposed in [68] is used along with an RFC physics simulator [66] (used to preserve physical stability in the generated poses) to eliminate artifacts in the synthesized motion sequences and thus render them physically plausible. On the other hand, motion debiasing compensates for biases produced due to dynamic mismatch during the imitation learning process, in order to help the model fully imitate motions. A dynamic mismatch [66] is the mismatch between the virtual character's pose and the goal (ground truth) motion [69].

Finally, in a rather recent work [70] in this category a model was implemented, that generates physically-realistic 3D pose sequences and the corresponding contact forces based on RGB videos instead of using mocap data. First, a CNN pose estimator is implemented to estimate both the 2D and 3D joint positions from each video frame. An Inverse Kinematics method is then used to transform the 3D joint positions for each frame into relative rotations of each body part of a skeleton model. Finally, an optimization technique is applied to minimize a loss function that combines a differentiable physics loss, a pose estimation loss, and a smoothness loss [70]. The model has the ability to physically correct noisy image-based pose estimations by incorporating physical parameters of motion and contact (e.g. contact forces) upon them. These optimized pose estimates can be used as a standard motion capture dataset to train a CVAE-based model that generates future motion sequences and contact forces.

4.3 Diversified Motion Synthesis

A recent line of work focuses on the synthesis of a large number of diversified high-quality motions with arbitrary duration, and visually-convincing variations in both space and time [71] [72] [73]. In [71] a DL-based generative model is developed, that can be used for both online and offline motion synthesis and control, producing a wide variety of realistic motions of any length. This model consists of an RNN (comprised of LSTM cells) for motion synthesis combined with an adversarial neural network (similar to a GAN) for "refining" the produced motion (control) so as to be identical to the reference input motion. A new deep learning model, called Dynamic Future Net (DFN) [72], was developed to produce diversified motion (walking-to-running, walking-to-dancing, etc.) with arbitrary duration, given a short-length pose sequence (e.g. walking). The DFN consists of 3 models: a pose encoder and a pose trajectory encoder, which embed pose and trajectory motion capture data into latent representations, respectively, and a stochastic latent RNN which learns the transition stochasticity of the past, current and future states in motions. In contrast to [37] [71], this network models explicitly the current state by relying on past and future states. Thus, this model is able to model both short-term and long-term randomness of motion. An interesting work [73] that covers both categories presented in this review, motion prediction and synthesis, as well as motion reconstruction, presents a Spatiotemporal Recurrent Network (STRNN), which enables long-term 3D skeletal human motion prediction and synthesis while learning diverse motion patterns. The STRNN incorporates a spatial, a temporal, and a residual network: the spatial network is a fully connected hierarchical neural network that segments the 3D human skeleton model in parts in order to model the spatial variations in a motion frame, without converging to the mean pose , while the temporal network is an LSTM-based RNN, which models the temporal variance between long-term motion sequences, without the need for supplementary temporal variables (e.g. using phase as in [44]). The residual network acts as a filter since it produces a filtered signal from the concatenation of the output of the temporal and spatial networks. In particular, this is a fully connected network that learns a signal which cancels high-frequency noise, such as periodic jumps, that are problematic in iterative motion prediction.

One of the most recent works in this application domain is [74], where a generative model called MoDi that produces diverse 3D pose sequences unconditionally, was introduced. The generator's architecture consists of two parts; the first part is a mapping network that constructs a latent space where the features of each motion are differentiated from the features of other motions, while the second part is a motion synthesis network that produces 3D pose sequences through skeleton-aware convolutions [51]. In supplementary, a set of motion style codes (standard deviation values) is introduced into the intermediate layers of the motion synthesis network to control the synthesis. This model is trained in an unsupervised way from unlabeled and unstructured motion capture data, which does not prevent the model from producing a robust latent space. Moreover, an inversion method was used to invert an unseen motion into the learned latent space of the generator in order to enable motion editing in the latent space [74]. It is worth mentioning another recent publication in diversified motion generation [75], which is a continuation of the work presented in [48]. In [75] a scene-aware hierarchical model that focuses on producing diversified goal-driven character-scene interactions. Similarly to [50], the model consists of three components each of which models a level of diversity; a CVAE that models the interaction diversity (e.g. interacting with various objects and performing different motion styles while doing so) by generating scene-agnostic motion sequences conditioned on a specific action and placing these poses into the given scene, a stochastic data-driven scene-conditioned path planning module based on A* algorithm to encode path diversity (e.g. produce different paths while reaching for a goal) and a transformer-based CVAE to synthesize diverse long body movements based on the calculated paths, scene context, character-object interactions, and actions [75].

5 Musculoskeletal Dynamics Estimation

As mentioned above deep learning techniques for musculoskeletal dynamics estimation can complement motion prediction and synthesis works by estimating biomechanical parameters that render motion estimation more accurate and realistic. For example, in [76] a deep learning method was developed for estimating foot contact mechanics and ground reaction forces in order to mitigate artifacts during motion synthesis. More specifically, a CNN was created to derive the distribution of vertical GRFs over the

feet from joint positions. Based on these estimations the foot contact labels can be extracted, which then can be used as kinematic constraints in an optimization-based IK method to eliminate footskating artifacts during motion estimation, model animations in uneven terrain or even performing image-based motion reconstruction [76]. It is worth mentioning that the authors in [76] created a human motion capture database enhanced with pressure insoles data, which can be used for both animation and simulation. Similarly, in [77], vertical, anterior-posterior, and medial-lateral GRFs during stair walking were estimated by a Bidirectional LSTM (an LSTM that has both forward and backward connections exploiting both past and future data) based on joint kinematics. Ground reaction forces can be used for both biomechanical analyses and therefore clinical applications as well as used in eliminating motion artifacts during simulation/animation as stated in [77].

Works such as [78], [79] are prime exemplars of combining musculoskeletal modeling with machine learning. In [78] a deep reinforcement learning-based architecture learns robust control policies to generate a variety of sophisticated human motions (e.g. dancing, kicking, etc.) using a 346-muscle-actuated model. The main advantage of this work is that a scalable algorithm is utilized that has the ability to control the movement of anatomically variant musculoskeletal models, which are fully- or under-actuated models (e.g. due to muscle weakness, prosthesis use, etc.). Furthermore, in one of the most recent works in this application domain [79] the gait of a full-body musculoskeletal model is controlled via a pre-trained reinforcement learning framework, named Generative GaitNet. GaitNet uses physics-based musculoskeletal simulation to produce walking sequences based on both anatomical (e.g. muscle deficits, body proportion, etc.) and gait conditions (like stride). In contrast with the work in [78], GaitNet learns a control policy with 608 muscle parameters. Deep reinforcement learning techniques were utilized to also simulate and control gait in modifying terrains [80] and in elderly patients, to explore how aging affects the kinesiology and muscle control during a fall [81]. Also, joint-actuation models were used instead of musculotendon ones to speed up computations and simplify modeling [82].

A state-of-the-art work that proposes an interesting turn in musculoskeletal dynamics estimation that leverages problems of both computational models (e.g. speed issues) and data-driven models (e.g. no modeling of the underlying physics that rule the internal body state of a human), is [7]. In [7] a deep learning model based on Physics-Informed Neural Networks (PINNs) for estimating muscle forces and joint angles of both lower and upper limbs given corresponding sEMG signals, was developed. PINNs combine conventional machine/deep learning models with physics laws, which are incorporated into the loss function in order to regularise the model's outputs. In this work, a CNN is utilized as the base deep learning technique to extract a feature space that represents the mapping between EMG signals and muscle forces and joint kinematics, while the total loss of the function of the framework is computed based on two losses: an MSE loss that minimizes the data prediction error and a physics-based loss that minimizes the equation of motion (physics law between muscle forces and joint motion). The latter loss can penalize the loss function of the CNN in order to render the model more robust and enhance its generalization ability. The evaluation results in terms of root mean square error and Pearson's correlation coefficient showed that this method performs better compared to other network architectures (e.g. CNN with more layers, SVR, etc.), since infusing physics-based domain knowledge in the network results in faster convergence speed [7]. A work similar to [7] is [8] where a Physics-Infused Neural Network (PIMNet a network that exploits both physics-based and ML methods) was developed for predicting joint torques and contact forces for each pose of a motion sequence. These predictions can be then utilized to produce more accurate short and long-term human motion predictions as it is claimed in [8].

In this category, there are also works with applications in computer vision, where physical forces in human-object interactions are predicted from visual information [83], [84]. In particular, in [83] a model consisting of an image feature extractor along with an encoder-decoder module was developed to predict contact points and forces between a human hand and an object given RGB videos and the initial state of the object. Then, the authors used a differentiable physics simulation mechanism to apply the inferred forces in a mesh object so as to accurately reproduce the actions obtained from video. By exploiting the gradients through this simulation, the model learns to optimize the simulated motion in terms of both contact points and force prediction. This joint optimization leads to increasing performance on both predictions and also the model extracts a physical representation that renders it capable of generalizing to new unseen objects using few training instances as showed in [83]. A previous work similar to [83] is [84], where an LSTM network with fully-connected layers was utilized to estimate the interaction force applied to an object while its shape is modified by an external load, based only on image information. Specifically, the model learns the mapping from image sequences depicting object shape alterations to interaction forces without relying on any force or torque sensors. This method can be also applied to estimate interaction forces against human skin or human limbs.

Other than works in computer graphics applications, there is also a plethora of works with biomechanical and medical applications, most of which focus on estimating knee contact forces (KCFs) [85], [86], [87], [88], [89] and muscle forces in the lower limbs [86], [88], [90], using various ML techniques. These works trained their surrogate models using datasets comprised of derivative data from musculoskeletal modeling analyses (Inverse Kinematics, Joint Reaction Analysis, etc.) based on motion marker tracking system measurements or signals from other sensors [87].

In [85] ANNs and support vector regression (SVR) were used for predicting medial and lateral knee contact forces (KCFs) during gait in real-time. The training of both models was performed either including ground reaction forces (GRFs) in the training dataset or not including GRFs, in order to investigate whether the omission of the (difficult to acquire) GRFs substantially affects the prediction power of the models. A work that also resembles the one in [85] utilizes ANN [87] for estimation of knee joint forces in sports movements (namely linear motions like gait and

TABLE 1
Overview of ML approaches for Musculoskeletal Dynamics Estimation[a]

Publication	Real-Time/Offline	Motion Dependent Variables	Type of Movement
(Giarmatzis, 2020) [85]	Real-time	With and without GRFs	Gait
(Burton, 2021) [86]	Offline/ Potential Real-time Use	With GRFs	Sit-to-Stand, left and right step down, and gait
(Stetter, 2019) [87]	Offline	With GRFs and IMU signals	Sports Movements (e.g. running, jumping etc.)
(Park, 2022) [79]	Real-time	No GRFs	Gait
(Rane, 2019) [88]	Real-time	GRFs and EMGs	Gait
(Zhu, 2020) [89]	Offline/ Potential Real-time Use	GRFs and EMGs	Gait
(Dao, 2019) [90]	Offline/ Potential Real-time Use	No GRFs	Gait
(Sohane, 2020) [91]	Offline	No GRFs	Squatting
(Zhang J., 2022) [7]	Offline	No GRFs	Walking and Wrist Motions
(Mourot, 2022) [76]	Offline	No GRFs	Walking, running, jumping, etc.
(Liu, 2022) [77]	Offline	No GRFs	Stair Walking
(Zhang Z., 2022) [8]	Offline	No GRFs	Various from H3.6m [92]
(Ehsani, 2020) [83]	Offline	No GRFs	Object Manipulation Motions
(Lee, 2019) [78]	Offline	No GRFs	Walking, Running, Jumping, Kicking, etc.

[a] The studies in Musculoskeletal Dynamics Estimation can be classified as follows: (i) studies that do or do not make use of motion-dependent variables such as GRFs, (ii) works that developed either real-time or offline frameworks and (iii) according to the targeted activity (e.g. gait, other daily life activities, etc.).

running including changes of direction and jumps) is performed based on mocap, force plate and data measured by wearable sensors. More analytically, knee kinematic and dynamic data from inertial measurement units (IMUs) were extracted. The model was trained using IMU measurements of all movements (as input) and the knee joint forces (as output) – which were obtained from musculoskeletal analysis considering only the stance phases of each motion – in order to find the correlation between them.

Another approach to KCF prediction was presented in [89], where a model that integrates the random forest (RF) with the artificial fish swarm algorithm was used. The model is using marker motion data, GRFs, KCFs and muscle electromyography (EMG) signals from patients with an instrumented knee replacement as input. The RF algorithm can handle high dimensional data and can be trained really fast, however, the optimization of its parameters can significantly improve its performance, hence, the artificial fish swarm algorithm was utilized to do so [89]. Another similar work is [91], where ML regression-based models were developed for estimating knee muscle force during squatting movements, given a variety of parameters such as joint angles, muscle forces, mass, and height. In particular, the authors in [91] developed four different frameworks, namely an RF model, a neural network, a generalized linear model (standard linear regression), and a decision tree, and compared them in terms of mean square error, Pearson's correlation coefficient and coefficient of determination (i.e. the square of Pearson's correlation coefficient) to conclude that the random forest one outperforms the others in a knee muscle force prediction scenario.

Concerning muscle force prediction, in [86] four different machine learning approaches were implemented to estimate the joint contact and muscle forces in patients with total knee replacement during a variety of everyday activities (sit-to-stand, left and right step down, and gait). The authors compared the performance of an RNN, a CNN, a fully-connected neural network, and principal component regression (i.e. a regression analysis based on PCA) to conclude that RNNs provide the most accurate predictions. The same conclusion was reached by the authors of [90], where a deep RNN (specifically an LSTM) was used to predict lower limb muscle forces from kinematic gait data since it can incorporate dynamic temporal relationships of the muscle forces. In order to increase the accuracy of the model's predictions they used a weight transfer learning technique, meaning that the weights from a pre-trained LSTM network model were stored and loaded into a new LSTM network model. The latter method is more beneficial when the available biomechanical data are scarce. Moreover, a CNN model was adopted in [88] to learn the mapping from kinematic space to force space. More specifically, the network predicted the medial knee joint reaction force, the forces for major muscle groups of the lower limb, and the EMG sensor measurements in real-time during gait. This model was validated in two different ways: by comparing the CNN's predictions with musculoskeletal modeling estimations and EMG sensor data, as well as ground truth tibiofemoral force data from the Grand Challenge Competition [93].

6 Comparative Analysis and Discussion

6.1 Motion Prediction

Regarding motion prediction, most deterministic methods consist of recurrent architectures, such as conventional Recurrent Neural Networks [22] and Gated Recurrent Units [19], which are suitable for solving problems where the input and/or output consist of sequences of points that are not correlated (e.g. when input and output are body joint trajectories) since they have the ability to model temporal dependencies. A work that combines RNN with GAN-based models, [40], shows the best performance assessed by the mean angle error, among studies using the same motion capture dataset, i.e. the Human 3.6m (H3.6m) by [92], followed by [27] and [32], according to Table 3.

On the contrary to CNN/RNN architectures that focus on exploring temporal dependencies in motion sequences,

TABLE 2
Publications since 2017 in Motion Prediction, Synthesis and Reconstruction[b]

Publication	Motion Prediction				Motion Synthesis				Motion Reconstruction	Source Code
	S-TMP	L-TMP	CMP	DMP	CMS	P-BMS	DMS	ST		
(Starke, 2019) [10]	✓	-	✓	-	✓	-	-	-	-	link
(Starke, 2020) [11]	✓	-	✓	-	✓	-	-	-	-	link
(Starke, 2021) [12]	✓	-	✓	-	✓	-	-	-	-	link
(Holden, 2017) [44]	✓	-	✓	-	✓	-	-	-	-	link
(Tang, 2018) [14]	✓	✓	-	-	-	-	-	-	-	-
(Martinez, 2017) [15]	✓	✓	-	-	-	-	-	-	-	link
(Liu, 2019) [16]	✓	✓	-	-	-	-	-	-	-	link
(Ma J., 2022) [62]	-	-	-	-	✓	-	✓	-	-	link
(Yasar, 2021) [19]	✓	✓	-	-	-	-	-	-	-	-
(Peng, 2018a) [94]	-	-	-	-	-	-	-	-	✓	link
(Gomes, 2020) [95]	-	-	-	-	-	-	-	-	✓	-
(Aberman, 2020) [96]	-	-	-	-	-	-	-	✓	-	link
(Bergamin, 2019) [67]	-	-	-	-	✓	✓	-	-	-	-
(Liu, 2021) [39]	✓	✓	✓	-	-	-	-	-	-	link
(Shi, 2020) [97]	-	-	-	-	-	-	-	-	✓	link
(Men, 2021) [53]	-	-	-	-	✓	-	-	-	-	-
(Harvey, 2020) [40]	✓	✓	✓	-	✓	-	-	-	-	-
(Henter, 2020) [37]	-	-	-	-	✓	-	-	-	-	link
(Ling, 2020) [52]	✓	-	✓	-	✓	-	-	-	-	link
(Lee, 2021) [65]	-	-	-	-	-	✓	-	-	-	link
(Peng, 2018b) [68]	-	-	-	-	-	✓	-	-	-	link
(Peng, 2021) [41]	-	-	-	-	-	✓	-	-	-	link
(Yuan, 2020) [66]	-	-	-	-	-	✓	-	-	✓	link
(Wang Z., 2019) [71]	✓	-	-	-	✓	-	✓	-	-	-
(Chen, 2020) [72]	✓	✓	-	-	-	-	✓	-	-	-
(Wang H., 2021) [73]	✓	✓	-	-	-	-	✓	-	✓	-
(Honda, 2020) [22]	✓	-	-	-	-	-	-	-	-	-
(Mao, 2020) [23]	✓	✓	-	-	-	-	-	-	-	link
(Cui, 2020) [25]	✓	✓	-	-	-	-	-	-	-	link
(Li M., 2020) [26]	✓	✓	-	-	-	-	-	-	-	link
(Zhang, 2021) [29]	✓	✓	-	✓	-	-	-	-	-	link
(Aliakbarian, 2020) [33]	✓	✓	-	✓	-	-	-	-	-	link
(Aliakbarian, 2021) [32]	✓	✓	-	✓	-	-	-	-	-	-
(Salzmann, 2022) [34]	✓	✓	-	✓	-	-	-	-	-	link
(Zhong, 2022) [27]	✓	✓	-	-	-	-	-	-	-	-
(Ma T., 2022) [28]	✓	✓	-	-	-	-	-	-	-	link
(Guo, 2022) [24]	✓	✓	-	-	-	-	-	-	-	link
(Ma H., 2022) [35]	✓	✓	-	✓	-	-	-	-	-	-
(Mao, 2021) [38]	-	-	✓	✓	-	-	-	-	-	link
(Pu, 2022) [46]	-	-	-	-	✓	-	-	-	-	-
(Aristidou, 2021) [45]	-	-	-	-	✓	-	✓	-	-	-
(Starke, 2022) [42]	✓	-	✓	-	✓	-	-	-	-	link
(Raab, 2022) [74]	-	-	-	-	-	-	✓	-	-	link
(Li J., 2021) [55]	-	-	-	-	-	-	✓	-	-	-
(Cai, 2021) [56]	-	-	-	-	-	-	✓	-	-	-
(Briq, 2022) [57]	-	-	-	-	-	-	✓	-	-	-
(Hassan, 2021) [50]	✓	-	✓	-	✓	-	-	-	-	link
(Wang J., 2021) [48]	✓	-	✓	-	✓	-	-	-	-	-
(Li P., 2022) [49]	-	-	-	-	✓	-	-	✓	-	link
(Wang J., 2022) [75]	-	-	-	-	✓	-	✓	-	-	-
(Xie, 2021) [70]	✓	-	-	-	-	✓	-	-	-	link
(Maeda, 2022) [69]	✓	-	-	-	✓	✓	-	-	-	link
(Mao, 2022) [36]	✓	✓	-	-	-	-	-	-	-	link
(Raab, 2023) [61]	-	-	-	-	✓	-	✓	-	-	link
(Mourot, 2020) [54]	-	-	-	-	✓	-	-	-	-	-
(Tevet, 2022) [58]	-	-	-	-	✓	-	✓	-	-	link
(Chang, 2022) [63]	-	-	-	-	✓	-	✓	✓	-	link
(Yuan, 2022) [64]	-	-	-	-	✓	-	✓	-	-	-

[b] The acronyms for each column are: (i) for Motion Prediction: S-TMP stands for Short-Term Motion Prediction, L-TMP for Long-Term Motion Prediction, CMP for Controlled Motion Prediction and DMP for Diverse Motion Prediction, (ii) for Motion Synthesis: CMS stands for Controlled Motion Synthesis, P-BMS for Physics-Based Motion Synthesis, DMS for Diversified Motion Synthesis, and ST for Style Transfer. Some papers address more than one application domains (thus adding the "Motion Reconstruction" column, even though this survey does not focus on this category), however, the initial classification was done based on which problem they focus on more. For example, [73] describes a framework that performs both motion prediction, synthesis, and reconstruction, yet the main contribution of this work is learning motion multi-modality. Moreover, we provide the publicly distributed source codes of the publications, if available.

many recent deterministic works, among which is the state-of-the-art publication [27] that achieves the second best performance (according to Table 3), are turning to graph-base models, Graph Neural Networks, and especially graph convolutional networks [23], [24], [25], [26], [28] to construct spatiotemporal feature spaces from historical human body pose sequences. These models offer a further understanding of the human internal body state by exploiting the human structure. Explanatory, in a graph neural network human 3D skeletons are represented as graphs (the nodes correspond to joints or body parts, and edges represent the relations between joints), and the network's main goal is to extract spatial dependencies between the joints of the pose skeletons in order to produce accurate and realistic future motion. Some of these works e.g. [23], [24], combine attention-based techniques with graph neural networks to guide the weights of the model in such a way that more accurate predictions are generated.

Furthermore, as for the probabilistic motion prediction, mostly Variational Autoencoders such as Conditional Variational Autoencoders [29], [32], [33], [35], [36] are utilized. In Table 3, the short- and long-term MAE results are reported also for probabilistic approaches such as [33], [34] and the one with the third best performance [32]. These results indicate from the set of diverse motions that such models can produce, there is at least one pose sequence that converges to the ground truth motion [33]. Thus, there is one solution of these stochastic models that is very close to a deterministic solution.

However, it is worth noting that the work in [16] (with the fifth-best performance) seems to have a few advantages over previous works. In particular, this work is presented as an alternative method to LSTM and GRU approaches [14], [15], which are unable to model long-term dependencies (long-term prediction) efficiently, as shown in Table 3. Moreover, in order to overcome strange distortions in the predicted motion, the authors developed a Lie algebra representation to explicitly encode the geometry and actual DoFs (degrees of freedom) of individual joints of 3D human skeletons. On the opposite, other studies, like [15], usually model the pose as 3D joint positions, thereby dealing with joints as independent entities and, thus, cannot capture intrinsic geometry.

Finally, open research questions are still in the application domain of ML-based motion prediction for synchronizing the locomotion of a rehabilitation exoskeleton limb according to the movement or muscle excitation of a natural limb [98]. This scientific area encompasses a plethora of publications where the motion of the exoskeleton is estimated using traditional (non-ML) methods, like the Gaussian process latent variable model [99] and human motion intent prediction–based control algorithm [100], whereas studies similar to [98] utilizing machine learning, have not been published in the last years.

6.2 Motion Synthesis

Most recent publications in motion synthesis utilize deterministic or probabilistic techniques, which are further categorized into physics-based methods, diversified motion synthesis, and style transfer techniques. Many studies combine RNNs with generative adversarial training. When RL or adversarial neural networks comprise the core mechanism for motion synthesis (given a specific scenario), the techniques are categorized as probabilistic, whereas if adversarial techniques are utilized for the creation of a pluralistic training set (e.g. set of scenes required to condition the movement trajectories) prior to deterministic model training, the methods are considered as deterministic. It is worth mentioning that even though adversarial techniques are used in deterministic motion generation —with most recent applications being synthesizing 3D character movement in interaction with a virtual scene [48] and producing movement from a single short motion sequence [49] —GANs are mainly present in probabilistic data-driven motion synthesis. Most deterministic motion synthesis methods rely on Recurrent Neural Networks [11], [12] (e.g. Long Short-Term Memory models [45]) and transformers with stacked attention layers [46].

This review also briefly presents studies on probabilistic motion synthesis [37], [50], [52], [55], [56], [57], [58], [61], [62], [63], [64]. Such works usually combine recurrent neural networks (simple RNNs, LSTMs, etc.) with adversarial neural networks [71], like GANs [40] or RL [52], [67] models, which learn control policies to generate the desired motion given a specific scenario. Other works [50], [55] in this application domain exploit variational autoencoders and specifically Conditional VAEs [50], [52], [56] since these models give the ability to produce multiple plausible motion sequences conditioned on past pose series or even parts of historical motions as in [56]. Moreover, in some of these works [56], [57] VAEs are a part of transformer-based architectures with multiple stack attention layers, which enable the extraction of both local and global long-term temporal features from 3D pose sequences and further synthesize motions of any lengths. Nevertheless, more recent works, published in late 2022 [58], [62], [63], [64] and early 2023 [61], explore the concept of diffusion models for motion generation conditioned on multiple data sources (text, audio, etc.). Diffusion models produce a distribution that can better express the many-to-many distribution matching problem (i.e. producing diverse motions) and thus increase the learning capacity of the neural network. This is an advantage over other probabilistic models, like VAEs, that imply a one-to-one mapping or produce a normal latent distribution that limits the stochasticity of the learning procedure [61].

Even though all of the aforementioned papers are quite recent and innovative, the architecture proposed in [37] slightly differentiates from the others because it can generate highly complex distributions. In addition, as the neural network is invertible, it gives the ability to directly calculate and maximize the likelihood of the data of the model during training, in contrast to GANs or VAEs. Moreover, MoGlow has built-in controllability, meaning that it can model conditional motion distributions without algorithmic latency, unlike VAEs [52]. Moreover, this model is task-agnostic which means that it is independent of restrictions like the anatomy of the animated character, or the motion being quasi-periodic (like in [44]).

The motion synthesis works presented previously use mostly motion capture data in order to train their models. Specifically, according to Table 4, most works create the training dataset by recording their own motion capture data sequences, since they try to produce quite specific move-

TABLE 3
Overview of ML approaches for short-term (400ms) and long-term (1000ms) prediction of motion on H3.6M[c]

Publication	ML Technique	Dataset	Short-term MAE	Long-term MAE
(Martinez, 2017) [15]	GRU	H3.6m [92]	1.23	1.89
(Tang, 2018) [14]	MHU	H3.6m	1.13	1.80
(Liu, 2019) [16]	RNN	H3.6m	0.93	1.36
(Mao, 2022) [36]	CVAE-based	NTU-RGB [101], GRAB [102], BABEL [103], HumanAct12 [60]	-	-
(Liu, 2021) [39]	GAN-based	H3.6M	1.43	1.75
(Harvey, 2020) [40]	RNN and GAN	H3.6M, LAFAN1 [40]	0.56 (320ms) and 0.64 (500ms)	0.79
(Honda, 2020) [22]	RNN	2D mocap data from videos	-	-
(Mao, 2020) [23]	GCN-based Feed-Forward	H3.6M, AMASS [104], 3DPW [105]	0.94	1.57
(Cui, 2020) [25]	GCN	H3.6M, CMU, 3DPW	0.86	1.20
(Li M., 2020) [26]	DMGNN	H3.6M, CMU	0.95	1.21
(Zhang, 2021) [29]	CVAE	H3.6M, CMU, AMASS, and more	-	-
(Aliakbarian, 2020) [33]	CVAE	H3.6M, CMU	0.68	1.03
(Aliakbarian, 2021) [32]	CVAE	H3.6M, CMU, etc.	0.65	1.02
(Salzmann, 2022) [34]	GRU-based	H3.6M, AMASS	1.01	1.63
(Zhong, 2022) [27]	GCN-based	H3.6M, AMASS, 3DPW	0.65	1.02
(Ma T., 2022) [28]	GCN-based	H3.6M, CMU, 3DPW	1.02	1.61
(Guo, 2022) [24]	GCN-based	Own Mocap Data (+Videos, 3D meshes etc.)	-	-
(Ma H., 2022) [35]	CVAE-based	H3.6M, HumanEva-I [106]	-	-
(Mao, 2021) [38]	Deep Generative	H3.6M, HumanEva-I	-	-

[c] The results of both short- and long-term prediction on the publicly available H3.6M dataset. The MAE (mean angle error) indicates the error (in degrees) between the reference joint angles (from the reference motion set) and the predicted ones and is computed across all activities (walking, running, jumping, etc. - 15 in total of H3.6M). The lower the MAE is, the better the accuracy that the model outputs. For [26] and [39] the short-term prediction MAE is the average of 11 actions of H3.6M, (namely, directions, greeting, phoning, posing, purchases, sitting, sitting down, taking photos, waiting, walking dog, walking together) and the long-term MAE is for the remaining 4 actions of the dataset (i.e. walking, eating, smoking, discussion). As for the [33] and [32] the short- and long-term forecasting results are averaged among the 4 actions of walking, eating, smoking, and discussing.

ments (e.g. martial art styles [12] or basketball dribbles [11]), which are hard-to-find in existing databases that contain one or few classes of action. Also, most motion synthesis publications offer real-time solutions, which can be used for automatic animation production and character control.

6.3 Musculoskeletal Dynamics Estimation

Most motion prediction and synthesis works that were described in the previous sections, provide human locomotion estimations relying only on kinematics data (e.g. joint angles) and neglect other physical parameters such as GRFs, joint, muscle, and contact forces, joint torques, etc. leading to unrealistic simulations. However, one can integrate deep learning models such as [76], [77] that estimate biomechanical parameters (e.g. foot contact mechanics or GRFs) in order to enhance the accuracy of motion estimation and provide more physically-plausible results. Other works in this category with applications in computer vision, like [83], can be used to infer physical forces from visual information (videos or images), which forces can be then exploited to enhance motion synthesis/prediction. Moreover, recent methods for musculoskeletal dynamics applications [7], [8] bring physics-based domain knowledge into the data-driven model (e.g. CNN, LSTM) as soft constraints (in the loss function). These Physics-Informed Neural Networks, while promoting physical consistency, do not explicitly model the complex underlying physics governing the human body (unlike FEM and traditional computational neuromusculoskeletal models), and, hence, they can be computationally fast. In both works [7], [8], it was shown that by penalizing/regularising the motion prediction/synthesis models, the robustness in the estimation of the forces and the kinematics improved. Lastly, in this category there are works such as [79] and [78] that synthesize the motion of muscle-actuated models via deep learning and pose as alternatives to musculoskeletal modeling software. It is worth mentioning that developing models for dynamics estimation with applications in visualization and graphics is still an open field for research.

Furthermore, there are many musculoskeletal dynamics works that are used to solve biomedical/medical problems. Such works can be used alongside motion estimation techniques to simulate the internal state of a 3D virtual character and take a turn toward physics-based animation. In this category, fall works that developed machine learning frameworks such as Artificial Neural Networks [85], [87], Random Forest [89], [91], Recurrent Neural Networks [86] and Convolutional Neural Networks [88] used for joint and muscle force estimation, especially for the lower limb.

6.4 Challenges and Research Aims

The use of machine learning in motion prediction and synthesis has shown a rise in the last three years (2020-2022), with over 40 works with new research directions reviewed in this work. As for the ML-based musculoskeletal dynamics estimation studies, which started to appear after 2018, we consider that more research in this application domain, investigating motion synthesis or motion prediction in combination with force prediction would be advantageous for multiple reasons. For biomedical engineering and clinical applications, body joint torques, joint force and muscle force estimation aids in the diagnosis or prognosis of a disease (e.g. osteoarthritis), while the availability of a model simulating the patient's body motion allows performing sensitivity and perturbation analysis for a different number of parameters, which is necessary for the design of subject-specific treatment (e.g. for total knee replacement surgery) and personalized intervention and rehabilitation strategies.

TABLE 4
Overview of ML approaches for motion synthesis[d]

Publication	ML Technique	Dataset	Real-Time/Offline
(Starke, 2019) [10]	NSM (auto-regression)	Own Mocap Dataset	Real-Time
(Starke, 2020) [11]	NSM with local motion phases	Own Mocap Dataset	Real-Time
(Starke, 2021) [12]	NSM with control modules	Own Mocap Dataset	Real-Time and Offline
(Henter, 2020) [37]	LSTM-based	CMU [107] and HDM05 [108] and dog Mocap from [43]	Real-time
(Peng, 2021) [41]	Adversarial RL	Commercial Mocap dataset	Real-time
(Starke, 2022) [42]	Periodic Autoencoder	Various Mocap Datasets (e.g. AIST++, dog mocap [43])	Real-time
(Aristidou, 2021) [45]	DL methods like LSTM	Own Mocap Dataset and AIST++	Real-time
(Pu, 2022) [46]	Transformer-based	AIST++ [59]	Offline
(Men, 2021) [53]	GAN (LSTM-based)	SBU [109] and HHOI [110] and 2C [111]	Real-Time
(Mourot, 2020) [54]	GAN-based	MPI-INF-3DHP [112]	Offline
(Wang J., 2021) [48]	GAN-based	PROX [113], GTA-IM [114]	Offline
(Li P., 2022) [49]	GAN-based	-	Offline
(Hassan, 2021) [50]	Autoregressive CVAE-based	Own Mocap dataset	Offline
(Ling, 2020) [52]	VAE + RL	Own Mocap Dataset	Real-time
(Li J., 2021) [55]	HM-VAE	AMASS, 3DPW, LAFAN1	Offline
(Cai, 2021) [56]	CVAE	H3.6M, CMU	Offline
(Briq, 2022) [57]	RTVAE-Multi	PROX, Charades [115]	Offline
(Raab, 2023) [61]	Transformer-based Diffusion Model	HumanML3D [116] and Mixamo [117] and more	Offline
(Ma J., 2022) [62]	Transformer-based Diffusion Model	AMASS [104], LAFAN1 [40], BABEL [103] and more	Offline
(Chang, 2022) [63]	Diffusion Model	dataset proposed by [118]	Offline
(Yuan, 2022) [64]	Transformer-based Diffusion Model	HumanML3D [116] and more	Offline
(Lee, 2021) [65]	Deep NN + RL	Public Available Datasets (like CMU [107] and Mixamo [117])	Real-time
(Yuan, 2020) [66]	RL-based	H3.6m	Real-time/Offline
(Bergamin, 2019) [67]	Motion Matching + RL	Own Mocap Dataset	Real-Time
(Peng, 2018b) [68]	RL	Commercial Mocap dataset or keyframed animations	Real-time
(Maeda, 2022) [69]	VAE + RL [68] + RL-based [66]	HDM05	Offline
(Xie, 2021) [70]	CVAE-based + CNN	H3.6M, HumanEva-I	Offline
(Wang Z., 2019) [71]	LSTM + GAN-based network	Own Mocap dataset + CMU	Real-time/Offline
(Chen, 2020) [72]	RNN	CMU	Offline
(Wang H., 2021) [73]	RNN	Various datasets like CMU, HDM05, H3.6m etc.	Offline
(Raab, 2022) [74]	GAN-based	Mixamo and HumanAct12	Offline
(Wang J., 2022) [75]	CVAE-based	PROX	Offline
(Tevet, 2022) [76]	Transformer-based Diffusion Model	HumanML3D [116] and more	Offline
(Aberman, 2020) [96]	GAN-based	Motion clips of an animated character and videos	Offline

[d] The 3D human full-body motion datasets mentioned in this table, were used for evaluating the cited works and are described thoroughly in Appendix B.

In computer graphics research on the other hand, such approaches may facilitate the creation of more realistic human movements in video games, adjusted to the estimated energy expenditure [119] and the character's long-term fatigue induced by selected actions [120]. Therefore, they open the path for modeling temporally evolving, ergonomic effects [121], which may allow in the future to synthesize fatigue-driven motion, and thereby produce more empathic computer graphics characters' animation.

Other open challenges for future research, are summarized below.

ML-based motion prediction

- Exploring novel or hybrid architectures that would produce more accurate predictions, i.e. resulting in small MAE as in [40].
- Evaluating existing models or developing new ones, using data from multiple action classes (beyond gait).
- Investigating new methods for eliminating distortions in the predicted movement, such as in [16].
- Using ML/DL methods to synchronize the locomotion of rehabilitation exoskeleton limbs based on the motion or muscle excitation of physical limbs [98], and generally focusing on developing motion prediction techniques for biomedical applications.

ML-based motion synthesis

- Targeting real-time algorithms for various applications, such as in the gaming industry.
- Using existing motion synthesis methods to create synthetic human movement databases (e.g. [40]).
- Focusing more on physics-based animation techniques that enhance the realistic feel and the accuracy of the synthetic movement.

ML-based musculoskeletal dynamics estimation

- Developing frameworks that estimate both the kinematics and dynamics of a digital avatar.
- Creating models that focus more on visualization and graphics applications (e.g. animating muscle-actuated models).
- Focusing more on architectures predicting joint or muscle forces in a short time horizon, instead of estimating them at the current frame.

7 Conclusion

Summarizing, this paper provides a thorough insight into state-of-the-art ML-based frameworks for motion prediction, synthesis, and musculoskeletal dynamics estimation problems. Such innovative works can accelerate the creation of more realistic animation, aid in robot planning for human-robot interaction applications, and significantly contribute to the design of exoskeletons or optimize the

treatment and rehabilitation of human (lower or upper) limb conditions. Furthermore, advanced motion generator schemes, that produce novel and more precise motions, can be used to bypass many crucial obstacles that both biomechanics and computer graphics engineers face, such as the hard-to-find motion capture data, as well as the scarcity of data obtained from human-object and human-human interaction. Further research is required to improve the generalization ability of the models to less frequent motion classes and transition events.

ACKNOWLEDGEMENT

This work has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No 871738 - CPSoSaware: Crosslayer cognitive optimization tools and methods for the lifecycle support of dependable CPSoS.

REFERENCES

[1] A. Seth, M. Sherman, J. Reinbolt, and S. Delp, "Opensim: a musculoskeletal modeling and simulation framework for in silico investigations and exchange," *Procedia IUTAM*, vol. 2, pp. 212–232, 2011.

[2] A. Seth, J. Hicks, T. Uchida, A. Habib, C. Dembia, J. Dunne, C. Ong, M. DeMers, A. Rajagopal, M. Millard, S. Hamner, E. Arnold, J. Yong, S. Lakshmikanth, M. Sherman, and S. Delp, "Opensim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement," *PLoS Computational Biology*, vol. 14, no. 7, p. e1006223, 2018.

[3] L. Shu, K. Yamamoto, J. Yao, P. Saraswat, Y. Liu, M. Mitsuishi, and N. Sugita, "A subject-specific finite element musculoskeletal framework for mechanics analysis of a total knee replacement," *Journal of Biomechanics*, vol. 77, pp. 146–154, 2018.

[4] I. Loi, S. Dimitar, and K. Moustakas, "Total knee replacement: Subject-specific modeling, finite element analysis, and evaluation of dynamic activities," *Frontiers in bioengineering and biotechnology*, vol. 9, p. 648356, 2021.

[5] A. Rudenko, L. Palmieri, M. Herman, K. Kitani, D. Gavrila, and K. Arras, "Human motion trajectory prediction: a survey," *The International Journal of Robotics Research*, vol. 39, p. 027836492091744, 2020.

[6] E. Halilaj, A. Rajagopal, M. Fiterau, J. Hicks, T. Hastie, and S. Delp, "Machine learning in human movement biomechanics: Best practices, common pitfalls, and new opportunities," *Journal of Biomechanics*, vol. 81, pp. 1–11, 2018.

[7] J. Zhang, Y. Zhao, F. Shone, Z. Li, A. F. Frangi, S. Xie, and Z. Zhang, "Physics-informed deep learning for musculoskeletal modelling: Predicting muscle forces and joint kinematics from surface emg," 2022. [Online]. Available: https://arxiv.org/abs/2207.01435

[8] Z. Zhang, Y. Zhu, R. Rai, and D. Doermann, "Pimnet: Physics-infused neural network for human motion prediction," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 8949–8955, 2022.

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

[10] S. Starke, H. Zhang, T. Komura, and J. Saito, "Neural state machine for character-scene interactions," *ACM Transactions on Graphics*, vol. 38, no. 6, p. 14, 2019.

[11] S. Starke, Y. Zhao, T. Komura, and K. Zaman, "Local motion phases for learning multi-contact character movements," *ACM Transactions on Graphics*, vol. 39, no. 4, p. 14, 2020.

[12] S. Starke, Y. Zhao, F. Zinno, and T. Komura, "Neural animation layering for synthesizing martial arts movements," *ACM Transactions on Graphics*, vol. 40, no. 4, p. 16, 2021.

[13] A. Rasouli, "Deep learning for vision-based prediction: A survey," 2020, arXiv:2007.00095.

[14] Y. Tang, L. Ma, W. Liu, and W.-S. Zheng, "Long-term human motion prediction by modeling motion context and enhancing motion dynamics," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18)*. AAAI Press, 2018, p. 935–941.

[15] J. Martinez, M. J. Black, and J. Romero, "On human motion prediction using recurrent neural networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4674–4683.

[16] Z. Liu, S. Wu, S. Jin, Q. Liu, S. Lu, R. Zimmermann, and L. Cheng, "Towards natural and accurate future motion prediction of humans and animals," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9996–10 004.

[17] X. Wang, Q. Chen, and W. Wang, "3d human motion editing and synthesis: A survey," *Computational and mathematical methods in medicine*, vol. 2014, p. 104535, 2014.

[18] O. Alemi and P. Pasquier, "Machine learning for data-driven movement generation: a review of the state of the art," 2019, arXiv:1903.08356.

[19] M. Yasar and T. Iqbal, "Improving human motion prediction through continual learning," 2021, arXiv:2107.00544.

[20] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1724–1734.

[21] F. Chollet, *Deep Learning with Python*. New York: Manning Publications, 2017.

[22] Y. Honda, R. Kawakami, and T. Naemura, "Rnn-based motion prediction in competitive fencing considering interaction between players," in *The 31st British Machine Vision Conference, BMVC*, 2020.

[23] W. Mao, M. Liu, and M. Salzmann, *History Repeats Itself: Human Motion Prediction via Motion Attention*, 11 2020, pp. 474–489.

[24] W. Guo, X. Bie, X. Alameda-Pineda, and F. Moreno, "Multi-person extreme motion prediction," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[25] Q. Cui, H. Sun, and F. Yang, "Learning dynamic relationships for 3d human motion prediction," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[26] M. Li, S. Chen, Y. Zhao, Y. Zhang, Y. Wang, and Q. Tian, "Dynamic multiscale graph neural networks for 3d skeleton based human motion prediction," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 211–220.

[27] C. Zhong, L. Hu, Z. Zhang, Y. Ye, and S. Xia, "Spatio-temporal gating-adjacency gcn for human motion prediction," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[28] T. Ma, Y. Nie, C. Long, Q. Zhang, and G. Li, "Progressively generating better initial guesses towards next stages for high-quality human motion prediction," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[29] Y. Zhang, M. Black, and S. Tang, "We are more than our joints: Predicting how 3d bodies move," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 3371–3381.

[30] D. Kingma and M. Welling, "Auto-encoding variational bayes," in *ICLR*, 2014.

[31] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *Foundations and Trends® in Machine Learning*, vol. 12, no. 4, pp. 307–392, 2019. [Online]. Available: https://doi.org/10.1561%2F2200000056

[32] S. Aliakbarian, F. Saleh, L. Petersson, S. Gould, and M. Salzmann, "Contextually plausible and diverse 3d human motion prediction," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 11 313–11 322.

[33] S. Aliakbarian, F. Saleh, M. Salzmann, L. Petersson, and S. Gould, "A stochastic conditioning scheme for diverse human motion prediction," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 5222–5231.

[34] T. Salzmann, M. Pavone, and M. Ryll, "Motron: Multimodal probabilistic human motion forecasting," 2022.

[35] H. Ma, J. Li, R. Hosseini, M. Tomizuka, and C. Choi, "Multi-objective diverse human motion prediction with knowledge distillation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2022 Oral)*, 2022.

[36] W. Mao, M. Liu, and M. Salzmann, "Weakly-supervised action transition learning for stochastic human motion prediction," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 8151–8160.

[37] G. Henter, S. Alexanderson, and J. Beskow, "Moglow: Probabilistic and controllable motion synthesis using normalising flows," *ACM Transactions on Graphics*, vol. 39, pp. 1–14, 2020.

[38] W. Mao, M. Liu, and M. Salzmann, "Generating smooth pose sequences for diverse human motion prediction," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 13 289–13 298.

[39] Z. Liu, K. Lyu, S. Wu, H. Chen, Y. Hao, and S. Ji, "Aggregated multi-gans for controlled 3d human motion prediction," 2021. [Online]. Available: https://arxiv.org/abs/2103.09755

[40] F. G. Harvey, M. Yurick, D. Nowrouzezahrai, and C. Pal, "Robust motion in-betweening," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 60–1, 2020.

[41] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa, "Amp: adversarial motion priors for stylized physics-based character control," *ACM Transactions on Graphics*, vol. 40, no. 4, pp. 1–20, 2021.

[42] S. Starke, I. Mason, and T. Komura, "Deepphase: periodic autoencoders for learning motion phase manifolds," *ACM Transactions on Graphics*, vol. 41, no. 4, pp. 1–13, 2022.

[43] H. Zhang, S. Starke, T. Komura, and J. Saito, "Mode-adaptive neural networks for quadruped motion control," *ACM Transactions on Graphics*, vol. 37, no. 4, pp. 1–11, 2018.

[44] D. Holden, T. Komura, and J. Saito, "Phase-functioned neural networks for character control," *ACM Transactions on Graphics*, vol. 36, no. 4, p. 13, 2017.

[45] A. Aristidou, A. Yiannakides, K. Aberman, D. Cohen-Or, A. Shamir, and Y. Chrysanthou, "Rhythm is a dancer: Music-driven motion synthesis with global structure," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2021.

[46] J. Pu and Y. Shan, "Music-driven dance regeneration with controllable key pose constraints," *CoRR*, vol. abs/2207.03682, 2022.

[47] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Advances in Neural Information Processing Systems*, vol. 3, 06 2014.

[48] J. Wang, S. Yan, B. Dai, and D. LIn, "Scene-aware generative network for human motion synthesis," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 12 201–12 210.

[49] P. Li, K. Aberman, Z. Zhang, R. Hanocka, and O. Sorkine-Hornung, "Ganimator: neural motion synthesis from a single sequence," *ACM Transactions on Graphics*, vol. 41, no. 4, pp. 1–12, 2022.

[50] M. Hassan, D. Ceylan, R. Villegas, J. Saito, J. Yang, Y. Zhou, and M. Black, "Stochastic scene-aware motion prediction," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 11 354–11 364.

[51] K. Aberman, P. Li, D. Lischinski, O. Sorkine-Hornung, D. Cohen-Or, and B. Chen, "Skeleton-aware networks for deep motion retargeting," *ACM Transactions on Graphics*, vol. 39, 2020.

[52] H. Ling, F. Zinno, G. Cheng, and M. Panne, "Character controllers using motion vaes," *ACM Transactions on Graphics*, vol. 39, 2020.

[53] Q. Men, H. Shum, E. Ho, and H. Leung, "Gan-based reactive motion synthesis with class-aware discriminators for human-human interaction," 2021.

[54] L. Mourot, F. L. Clerc, C. Thébault, and P. Hellier, "Jumps: Joints upsampling method for pose sequences," 2020. [Online]. Available: https://arxiv.org/abs/2007.01151

[55] J. Li, R. Villegas, D. Ceylan, J. Yang, Z. Kuang, H. Li, and Y. Zhao, "Task-generic hierarchical human motion prior using vaes," in *2021 International Conference on 3D Vision (3DV)*, 2021, pp. 771–781.

[56] Y. Cai, Y. Wang, Y. Zhu, T.-J. Cham, J. Cai, J. Yuan, J. Liu, C. Zheng, S. Yan, H. Ding, X. Shen, D. Liu, and N. Thalmann, "A unified 3d human motion synthesis model via conditional variational auto-encoder," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 11 625–11 635.

[57] R. Briq, C. Zou, L. Pishchulin, C. Broaddus, and J. Gall, "Recurrent transformer variational autoencoders for multi-action motion synthesis," 2022. [Online]. Available: https://arxiv.org/abs/2206.06741

[58] G. Tevet, S. Raab, B. Gordon, Y. Shafir, D. Cohen-Or, and A. H. Bermano, "Human motion diffusion model," 2022. [Online]. Available: https://arxiv.org/abs/2209.14916

[59] R. Li, S. Yang, D. Ross, and A. Kanazawa, "Ai choreographer: Music conditioned 3d dance generation with aist++," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 13 381–13 392.

[60] C. Guo, X. Zuo, S. Wang, S. Zou, Q. Sun, A. Deng, M. Gong, and L. Cheng, "Action2motion: Conditioned generation of 3d human motions," in *In Proceedings of the 28th ACM International Conference on Multimedia*, 2020, p. 2021–2029.

[61] S. Raab, I. Leibovitch, G. Tevet, M. Arar, A. H. Bermano, and D. Cohen-Or, "Single motion diffusion," 2023. [Online]. Available: https://arxiv.org/abs/2302.05905

[62] J. Ma, S. Bai, and C. Zhou, "Pretrained diffusion models for unified human motion synthesis," 2022. [Online]. Available: https://arxiv.org/abs/2212.02837

[63] Z. Chang, E. J. C. Findlay, H. Zhang, and H. P. H. Shum, "Unifying human motion synthesis and style transfer with denoising diffusion probabilistic models," 2022. [Online]. Available: https://arxiv.org/abs/2212.08526

[64] Y. Yuan, J. Song, U. Iqbal, A. Vahdat, and J. Kautz, "Physdiff: Physics-guided human motion diffusion model," 2022. [Online]. Available: https://arxiv.org/abs/2212.02500

[65] S. Lee, S. Lee, Y. Lee, and J. Lee, "Learning a family of motor skills from a single motion clip," *ACM Transactions on Graphics*, vol. 40, no. 4, pp. 1–13, 2021.

[66] Y. Yuan and K. Kitani, "Residual force control for agile human behavior imitation and extended motion synthesis," in *Advances in Neural Information Processing Systems*, 2020.

[67] K. Bergamin, S. Clavet, D. Holden, and J. Forbes, "Drecon: data-driven responsive control of physics-based characters," *ACM Transactions on Graphics*, vol. 38, pp. 1–11, 2019.

[68] X. Peng, P. Abbeel, S. Levine, and M. Panne, "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills," *ACM Transactions on Graphics*, vol. 37, 2018.

[69] T. Maeda and N. Ukita, "Motionaug: Augmentation with physical correction for human motion prediction," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[70] K. Xie, T. Wang, U. Iqbal, Y. Guo, S. Fidler, and F. Shkurti, "Physics-based human motion estimation and synthesis from videos," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 11 512–11 521.

[71] Z. Wang, J. Chai, and S. Xia, "Combining recurrent neural networks and adversarial training for human motion synthesis and control," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 1, pp. 14–28, 2019.

[72] W. Chen, H. Wang, Y. Yuan, T. Shao, and K. Zhou, "Dynamic future net: Diversified human motion generation," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, p. 2131–2139.

[73] H. Wang, E. Ho, H. Shum, and Z. Zhu, "Spatiotemporal manifold learning for human motions via long-horizon modeling," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 1, pp. 216 –227, 2021.

[74] S. Raab, I. Leibovitch, P. Li, K. Aberman, O. Sorkine-Hornung, and D. Cohen-Or, "Modi: Unconditional motion synthesis from diverse data," 2022. [Online]. Available: https://arxiv.org/abs/2206.08010

[75] J. Wang, Y. Rong, J. Liu, S. Yan, D. Lin, and B. Dai, "Towards diverse and natural scene-aware 3d human motion synthesis," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[76] L. Mourot, L. Hoyet, F. L. Clerc, and P. Hellier, "Underpressure: Deep learning for foot contact detection, ground reaction force estimation and footskate cleanup," 2022. [Online]. Available: https://arxiv.org/abs/2208.04598

[77] D. Liu, M. He, M. Hou, and Y. Ma, "Deep learning based ground reaction force estimation for stair walking using kinematic data," *Measurement*, vol. 198, p. 111344, 2022.

[78] S. Lee, M. Park, K. Lee, and J. Lee, "Scalable muscle-actuated human simulation and control," *ACM Transactions on Graphics*, vol. 38, no. 4, p. 1–13, 2019.

[79] J. Park, S. Min, P. S. Chang, J. Lee, M. Park, and J. Lee, "Generative gaitnet," 2022. [Online]. Available: https://arxiv.org/abs/2201.12044

[80] J. Wang, W. Qin, and L. Sun, "Terrain adaptive walking of biped neuromuscular virtual human using deep reinforcement learning," *IEEE Access*, vol. 7, pp. 92 465–92 475, 2019.

[81] K. Nowakowski, K. Kirat, and T.-T. Dao, "Deep reinforcement learning coupled with musculoskeletal modelling for a better

understanding of elderly falls," *Medical and Biological Engineering and Computing*, vol. 60, no. 6, pp. 1745–1761, 2022.

[82] Y. Jiang, T. V. Wouwe, F. D. Groote, and C. K. Liu, "Synthesis of biologically realistic human motion using joint torque actuation," *ACM Transactions on Graphics*, vol. 38, no. 4, pp. 1–12, jul 2019. [Online]. Available: https://doi.org/10.1145%2F3306346.3322966

[83] K. Ehsani, S. Tulsiani, S. Gupta, A. Farhadi, and A. Gupta, "Use the force, luke! learning to predict physical forces by simulating effects," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 221–230.

[84] L. S.-C. Hwang Wonjun, "Inferring interaction force from visual information without using physical force sensors," *Sensors (Basel)*, vol. 17, no. 11, p. 2455, 2017.

[85] G. Giarmatzis, E. I. Zacharaki, and K. Moustakas, "Real-time prediction of joint forces by motion capture and machine learning," *Sensors*, vol. 20, no. 23, p. 6933, 2020.

[86] W. S. Burton, C. A. Myers, and P. J. Rullkoetter, "Machine learning for rapid estimation of lower extremity muscle and joint loading during activities of daily living," *Journal of Biomechanics*, vol. 123, p. 110439, 2021.

[87] B. Stetter, S. Ringhof, F. Krafft, S. Sell, and T. Stein, "Estimation of knee joint forces in sport movements using wearable sensors and machine learning," *Sensors*, vol. 19, no. 17, p. 3690, 2019.

[88] L. Rane, Z. Ding, A. Mcgregor, and A. Bull, "Deep learning for musculoskeletal force prediction," *Annals of Biomedical Engineering*, vol. 47, no. 3, p. 778–789, 2019.

[89] Y. Zhu, W. XU, G. Luo, H. Wang, Y. Jingjing, and W. Lu, "Random forest enhancement using improved artificial fish swarm for the medial knee contact force prediction," *Artificial Intelligence in Medicine*, vol. 103, p. 101811, 2020.

[90] T. T. Dao, "From deep learning to transfer learning for the prediction of skeletal muscle forces," *Medical and Biological Engineering and Computing*, vol. 57, no. 5, pp. 1049–1058, 2019.

[91] A. Sohane and R. Agarwal, "Knee muscle force estimating model using machine learning approach," *The Computer Journal*, 2020.

[92] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 7, 2013.

[93] B. Fregly, T. Besier, D. Lloyd, S. Delp, S. Banks, M. Pandy, and D. D'Lima, "Grand challenge competition to predict in vivo knee loads," *Journal of Orthopaedic Research*, vol. 30, no. 4, p. 503–513, 2012.

[94] X. Peng, A. Kanazawa, J. Malik, P. Abbeel, and S. Levine, "Sfv:reinforcement learning of physical skills from videos," *ACM Transactions on Graphics*, vol. 37, pp. 1–14, 2018.

[95] T. Gomes, R. Martins, J. Ferreira, and E. Nascimento, "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)*, 2020, pp. 3355–3364.

[96] K. Aberman, Y. Weng, D. Lischinski, D. Cohen-Or, and B. Chen, "Unpaired motion style transfer from video to animation," *ACM Transactions on Graphics*, vol. 39, no. 4, p. 12, 2020.

[97] M. Shi, K. Aberman, A. Aristidou, T. Komura, D. Lischinski, D. Cohen-Or, and B. Chen, "Motionet: 3d human motion reconstruction from monocular video with skeleton consistency," *ACM Transactions on Graphics*, vol. 40, 2020.

[98] J.-L. Ren, Y.-H. Chien, E.-Y. Chia, L.-C. Fu, and J.-S. Lai, "Deep learning based motion prediction for exoskeleton robot control in upper limb rehabilitation," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 5076–5082.

[99] X. Jin, J. Guo, Z. Li, and R. Wang, "Motion prediction of human wearing powered exoskeleton," *Mathematical Problems in Engineering*, vol. 2020, p. 8, 2020.

[100] J. Yang and C. Peng, "Adaptive motion intent understanding–based control of human–exoskeleton system," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 235, no. 2, pp. 180–189, 2021.

[101] A. Shahroudy, J. Liu, T. Ng, and G. Wang, "Ntu rgb+d: A large scale dataset for 3d human activity analysis," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[102] O. Taheri, N. Ghorbani, M. Black, and D. Tzionas, "Grab: A dataset of whole-body human grasping of objects," in *Computer Vision – ECCV 2020*, 2020, pp. 581–600.

[103] A. Punnakkal, A. Chandrasekaran, N. Athanasiou, M. A. Quirós Ramírez, and M. Black, "Babel: Bodies, action and behavior with english labels," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 722–731.

[104] N. Mahmood, N. Ghorbani, N. Troje, G. Pons-Moll, and M. Black, "Amass: Archive of motion capture as surface shapes," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 5441–5450.

[105] T. Marcard, R. Henschel, M. Black, B. Rosenhahn, and G. Pons-Moll, "Recovering accurate 3d human pose in the wild using imus and a moving camera," in *Computer Vision – 15th European Conference on Computer Vision (ECCV) 2018*, 2018, pp. 614–631.

[106] L. Sigal, A. Balan, and M. Black, "Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion," *International Journal of Computer Vision*, vol. 87, no. 1-2, pp. 4–27, 2010.

[107] "Cmu graphics lab. 2003. carnegie mellon university motion capture database." http://mocap.cs.cmu.edu/.

[108] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber, "Documentation mocap database hdm05. technical report cg-2007-2. universität bonn, bonn, germany." $http://resources.mpi-inf.mpg.de/HDM05/07_MuRoClEbKrWe_HDM05.pdf$, 2007.

[109] K. Yun, J. Honorio, D. Chattopadhyay, T. Berg, and D. Samaras, "Two-person interaction detection using body-pose features and multiple instance learning," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2012, pp. 28–35.

[110] T. Shu, M. Ryoo, and S.-C. Zhu, "Learning social affordance for human-robot interaction," in *The 25th International Joint Conference on Artificial Intelligence (IJCAI)*, 2016, p. 3454–3461.

[111] Y. Shen, L. Yang, E. Ho, and H. Shum, "Interaction-based human activity comparison," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 8, p. 2620–2633, 2019.

[112] D. Mehta, H. Rhodin, D. Casas, P. Fua, O. Sotnychenko, W. Xu, and C. Theobalt, "Monocular 3d human pose estimation in the wild using improved cnn supervision," in *2017 International Conference on 3D Vision (3DV)*, 2017, pp. 506–516.

[113] M. Hassan, V. Choutas, D. Tzionas, and M. Black, "Resolving 3d human pose ambiguities with 3d scene constraints," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 2282–2292.

[114] Z. Cao, H. Gao, K. Mangalam, Q.-Z. Cai, M. Vo, and J. Malik, *Long-Term Human Motion Prediction with Scene Context*, 2020, pp. 387–404.

[115] G. A. Sigurdsson, A. Gupta, C. Schmid, A. Farhadi, and K. Alahari, "Charades-ego: A large-scale dataset of paired third and first person videos," 2018. [Online]. Available: https://arxiv.org/abs/1804.09626

[116] C. Guo, S. Zou, X. Zuo, S. Wang, W. Ji, X. Li, and L. Cheng, "Generating diverse and natural 3d human motions from text," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 5142–5151.

[117] "Adobe systems incs. 2018. mixamo." https://www.mixamo.com.

[118] S. Xia, C. Wang, J. Chai, and J. Hodgins, "Realtime style transfer for unlabeled heterogeneous human motion," *ACM Transactions on Graphics*, vol. 34, no. 4, p. 1–10, 2015.

[119] K. Risvas, M. Pavlou, E. I. Zacharaki, and K. Moustakas, "Biophysics-based simulation of virtual human model interactions in 3d virtual scenes," in *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, 2020, pp. 119–124.

[120] N. Cheema, L. Frey-Law, K. Naderi, J. Lehtinen, P. Slusallek, and P. Hämäläinen, "Predicting mid-air interaction movements and fatigue using deep reinforcement learning," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems CHI '20*, 2020, p. 1–13.

[121] M. Pavlou, D. Laskos, E. I. Zacharaki, K. Risvas, and K. Moustakas, "Xrsise: an xr training system for interactive simulation and ergonomics assessment," *Frontiers in Virtual Reality*, vol. 2, p. 17, 2021.