Edinburgh Research Explorer

MotioNet: 3D Human Motion Reconstruction from Video with Skeleton Consistency

MotioNet: 3D Human Motion Reconstruction from Monocular Video with Skeleton Consistency

MINGYI SHI, Shandong University and AICFVE, Beijing Film Academy

KFIR ABERMAN, AICFVE, Beijing Film Academy and Tel-Aviv University

ANDREAS ARISTIDOU, University of Cyprus and RISE Research Centre

TAKU KOMURA, Edinburgh University

DANI LISCHINSKI, Shandong University, The Hebrew University of Jerusalem, and AICFVE, Beijing Film Academy

DANIEL COHEN-OR, Tel-Aviv University and AICFVE, Beijing Film Academy

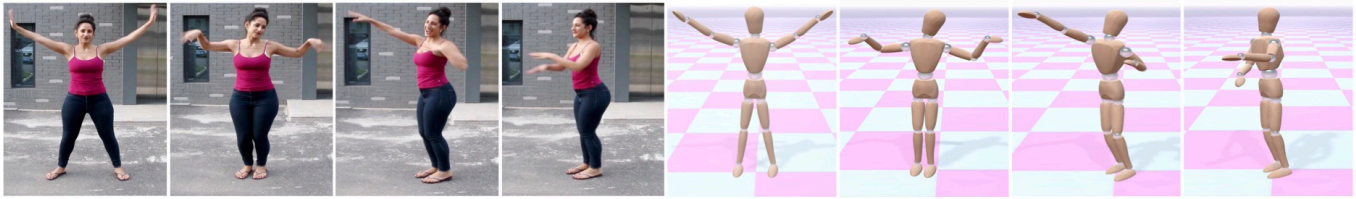BAOQUAN CHEN, CFCS, Peking University and AICFVE, Beijing Film Academy

Fig. 1. Given a monocular video of a performer, our approach, MotioNet, reconstructs a complete representation of the motion, consisting of a single symmetric skeleton, and a sequence of global root positions and 3D joint rotations. Thus, inverse kinematics is effectively integrated within the network, and is data-driven, rather than based on a universal prior. The images on the right were rendered from the output of our system after a simple rigging process.

We introduce *MotioNet*, a deep neural network that directly reconstructs the motion of a 3D human skeleton from monocular video. While previous methods rely on either rigging or inverse kinematics (IK) to associate a consistent skeleton with temporally coherent joint rotations, our method is the first data-driven approach that directly outputs a kinematic skeleton, which is a complete, commonly used, motion representation. At the crux of our approach lies a deep neural network with embedded kinematic priors, which decomposes sequences of 2D joint positions into two separate attributes: a single, symmetric, skeleton, encoded by bone lengths, and a sequence of 3D joint rotations associated with global root positions and foot contact labels. These attributes are fed into an integrated forward kinematics (FK) layer that outputs 3D positions, which are compared to a ground truth. In addition, an adversarial loss is applied to the velocities of the recovered rotations, to ensure that they lie on the manifold of natural joint rotations. The key advantage of our approach is that it learns to infer natural joint rotations directly from the training data, rather than assuming an underlying model, or inferring them from joint positions using a data-agnostic IK solver. We show that enforcing a single consistent skeleton along with temporally coherent joint rotations constrains the solution space, leading to a more robust handling of self-occlusions and depth ambiguities.

CCS Concepts: • **Computing methodologies → Motion processing**; **Neural networks**.

Authors' addresses: Mingyi Shi, Shandong University, AICFVE, Beijing Film Academy; Kfir Aberman, AICFVE, Beijing Film Academy, Tel-Aviv University; Andreas Aristidou, University of Cyprus, 75, Kallipoleos, Nicosia, Cyprus, 1678, RISE Research Centre, a.aristidou@ieee.org; Taku Komura, Edinburgh University; Dani Lischinski, Shandong University, The Hebrew University of Jerusalem, AICFVE, Beijing Film Academy; Daniel Cohen-Or, Tel-Aviv University, AICFVE, Beijing Film Academy; Baoquan Chen, CFCS, Peking University, AICFVE, Beijing Film Academy.

Additional Key Words and Phrases: Pose estimation, motion capturing, motion analysis

1 INTRODUCTION

Capturing the motion of humans has long been a fundamental task with a wide spectrum of applications in data-driven computer animation, special effects, gaming, activity recognition, and behavioral analysis. Motion is most accurately captured in a controlled setting using specialized hardware, such as magnetic trackers, depth sensors, or multi-camera optical systems. An alternative approach that has been researched extensively in recent years is to perform pose estimation and 3D motion reconstruction from ordinary monocular RGB video.

Motion capture from monocular video offers many advantages, such as a simple uncontrolled setup, low cost, and a non-intrusive capture process. While 3D human pose estimation is highly challenging due to depth ambiguities and occlusions, significant progress has been achieved in recent years by data-driven learning-based approaches. These approaches utilize deep neural networks to learn strong priors about the expected motion, which can significantly help with disambiguation and completion of missing data.

Given a video recording of a human motion, our ultimate goal is to reconstruct the motion in 3D space. One family of existing methods extract a sequence of 3D poses from the video, where each

pose is specified by the 3D location of each joint. However, while the resulting representation may suffice for some applications, it is *incomplete*. In particular, it does not contain all the information necessary to drive a rigged and skinned virtual 3D character, and the temporal consistency of the skeleton's bone lengths is not guaranteed. While joint rotations may be recovered from joint positions via inverse kinematics (IK), the solution is generally not unique, as demonstrated in Figure 2. Furthermore, enforcing soft temporal coherence constraints over per-frame pose estimations may not ensure that the skeleton geometry remains invariant across all frames, and might result in unnatural movements.

Another group of works is aiming at the recovery of a parametric model that depicts the geometry of the body including joint rotations (see Section 2). However, further rigging is required in order to extract a kinematic skeleton from such a model.

In this paper, we introduce *MotioNet*, a deep neural network, trained to reconstruct the motion of a single performer from an ordinary monocular video (Figure 1). Instead of inferring a sequence of 3D joint positions, our network learns to extract a sequence of *3D joint rotations* applied to a *single* 3D skeleton. Thus, IK is effectively integrated within the network, and, consequently, is data-driven (learned). Enforcing both a single skeleton and temporally coherent joint rotations not only constrains the solution space, ensuring consistency, but also leads to a more robust handling of self-occlusions and depth ambiguities.

To train our network, we leverage existing datasets that contain accurately captured full 3D human motions. Sequences of 3D poses are projected into 2D, and the network learns to decompose the resulting 2D joint position sequences into two separate attributes: a single, symmetric, skeleton, encoded by bone lengths, which define a geometric invariant along the entire sequence, and a sequence of 3D joint rotations, which capture the dynamic aspect of the motion. The 3D skeleton and joint rotations are fed into an integrated forward kinematics (FK) layer which applies the rotations successively along the bone hierarchy to reconstruct the original 3D motion sequence. In addition to the above, our network predicts a sequence of global positions of the root joint, as well as foot contact labels, because of the perceptual importance of the latter.

The network loss is a combination of terms that account for the bone lengths of the 3D skeleton, root global positions and foot contact labels, as well as joint positions recovered by the FK layer. While these attributes are compared to the ground truth 3D motion, the joint rotations are learned using an adversarial loss, which encourages their velocities to have a distribution of natural rotations. In addition, in order to mitigate foot skating artifacts, we add a foot contact loss, to encourage the velocity of each foot to be zero in frames where it should be in contact with the ground.

A key advantage of our approach is that it does not require an IK step, which is data-agnostic and assumes an underlying constrained model. Instead, the task is integrated into the network, which learns to infer joint rotations directly from training data of real human motions, rather than solving for them. Furthermore, as our system represents motion in the space of temporal convolutional filters, the learned motions are naturally smooth. All of this leads to a more data-driven human motion reconstruction, via the FK layer.
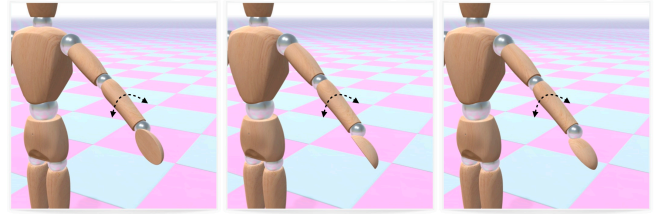

Fig. 2. Joint rotation ambiguity. Given a set of fixed 3D joint positions, multiple limb rotations can connect every pair of consecutive joints. Thus, recovered 3D joint positions alone are not sufficient for driving a rigged and skinned virtual 3D character.

In order to bridge the gap between the training data and videos in the wild, we inject joint positional noise to the training input sequences and augment them with confidence values whose distribution mimics that of confidence values extracted by [Cao et al. 2018] from a variety of real videos. This augmentation step constitutes a regularizer in the space of the solutions, improves the stability of the results, and increases the robustness to occlusions. An extensive set of experiments and ablation studies that we conducted to study the performance of our system and its different components, demonstrate the quality and stability of our end-to-end, fully data-driven approach for monocular motion extraction.

2 RELATED WORK

Prior to the deep learning era, various works performed 3D human pose estimation by performing exemplar-based IK to a set of constraints such as 2D keypoints [Grochow et al. 2004], or by fitting a skeleton to a voxel-based 3D model that was extracted over time from a video, e.g., [Liu et al. 2013; Vlasic et al. 2008]. The latter approaches are broadly divided into generative methods, which reconstruct human pose by fitting a template model to the observed data [Wei et al. 2012; Ye and Yang 2014]; discriminative methods, which cluster pixels to hypothesize body joint positions, fit a model, and then track the skeleton [Sharp 2012; Shotton et al. 2013]; and hybrid methods, which combine the aforementioned techniques to achieve higher accuracy [Baak et al. 2011]. See Sarafianos et al. [2016] for a detailed overview of 3D pose estimation methods.

3D Joint Position Reconstruction with Deep Learning: Recently, enormous amount of work has been devoted to the development of deep learning networks that extract the 3D joint positions of the human body from an image or a video sequence; Some methods, which extract the poses directly from the image pixels, have been popular due to the availability of video datasets that are associated with ground truth motion capture data (e.g., Human3.6M [Ionescu et al. 2014] and HumanEva [Sigal et al. 2009]). These methods use supervised learning to estimate the 3D joint positions [Pavlakos et al. 2017; Tekin et al. 2016; Tome et al. 2017; Toshev and Szegedy 2014; Zhou et al. 2016]. However, these 3D human pose estimation methods do not employ rigid body constraints. Thus, some works integrate an iterative error feedback to regress bones and improve 3D pose estimation [Carreira et al. 2016; Liang et al. 2018; Luvizon et al. 2018]. The main issue with such methods is that training data

is captured and annotated in a controlled environment, thus they do not generalize well to the real world. To address this issue, some works use weak supervision, where both motion capture data and images in the wild are used [Mehta et al. 2017a]. In addition, most of these works operate on a per-frame basis, which might lead to temporal incoherence and oscillations in motion. There have been efforts in exploiting temporal information from video to produce more robust predictions [Katircioglu et al. 2018; Tekin et al. 2016], e.g., using long short-term memory (LSTM) networks [Lee et al. 2018a; Lin et al. 2017].

Another family of 3D pose estimators relies on 2D pose estimators by first predicting 2D key joint positions in image space [Cao et al. 2018; Chen et al. 2018a; Newell et al. 2016; Papandreou et al. 2017; Wei et al. 2016], which are subsequently lifted to 3D either by regression [Martinez et al. 2017; Moreno-Noguer 2017] or model fitting [Chen and Ramanan 2017]. The latter methods, which are the most popular, are benefited by the availability of large-scale datasets and perform a k-nearest neighbor search for a predicted set of 2D keypoints over a dictionary of 2D skeletons for which the 3D skeleton is available, and then simply output the corresponding 3D pose [Tekin et al. 2017; Yiannakides et al. 2019; Zhou et al. 2017]. Some extensions integrate joint constraints to overcome the ambiguity in transferring 2D human poses into their 3D pose representations [Akhter and Black 2015; Mehta et al. 2019], enforce priors about bone lengths, and projection consistency with the 2D ground truth [Brau and Jiang 2016; Dabral et al. 2018; Ramakrishna et al. 2012; Wang et al. 2019], or use multi-view cameras to improve supervision [Rhodin et al. 2018b]. Others use adversarial training to generate anthropometrically feasible poses [Yang et al. 2018]. In general, these methods outperform those based on direct estimation, since they benefit from intermediate supervision, which can be trained on images in the wild. Indeed, the state-of-the-art results for 3D pose estimation are achieved by a recent two-stage method of Pavllo et al. [2019].

However, a major limitation of these pose estimation methods, is that the positional representation can only describe a small set of the full human motion articulation, which highly complex and dynamic. Moreover, since motion is not assigned to a character model to enforce kinematic constraints, resulting temporal bone length may be violated. Some works use reference objects in the scene with known position and scale, or other distinct parts of the human body (e.g., heads), to better estimate the person's height [Bieler et al. 2019; Günel et al. 2018]. More recently, body regression models have been employed to enforce some form of skeleton constraints [Mehta et al. 2019; Yoshiyasu et al. 2018]. Our goal is to enable a complete representation of motion, which consists of a skeleton and joint rotations.

In addition, positional representation allows ambiguities regarding the geometry and orientation of the limb segments as illustrated in Figure 2). A full motion representation requires reconstruction of joint rotations. Converting a positional representation into 3D joint rotations is rather challenging, since this problem is highly underconstrained [Dong et al. 2017]. Mehta et al. [2017b] fit a rigged skeleton model over time to estimated 2D and 3D joint locations, and use IK as a post-process to recover 3D rotations of each joint. In contrast, our network directly extracts 3D joint rotations from 2D

joint positions, producing the same format as that of commercial motion capture systems. The sequence of 3D joint rotations refers to a single 3D skeleton, thus ensuring that bone lengths remain constant over time, and learning joint quaternions directly from the data results in a natural motion without assuming an underlying model or employing IK solvers.

3D shape recovery: A parallel, related line of research aims at recovering full 3D shapes of people from images [Huang et al. 2017; Kolotouros et al. 2019a], using mainly statistical body shape (SMPL) models [Loper et al. 2015]. These approaches are typically based on parameterized 3D meshes and regression, and focus less on pose accuracy [Anguelov et al. 2005; Loper et al. 2015]. More recently, a number of deep convolutional methods have been proposed that map all human pixels of an RGB image to the 3D surface of a human body [Alldieck et al. 2018; Güler et al. 2018; Lassner et al. 2017; Zheng et al. 2019]. Next, in order to extract the 3D human pose from silhouettes, they fit a body shape model to the 2D joints, and minimize an objective function that penalizes the error between the projected 3D model joints and detected 2D joints [Kanazawa et al. 2018; Kolotouros et al. 2019b; Pavlakos et al. 2018b; Zhou et al. 2018]. As a consequence, they predict the joint rotations as a function of the surface. Moreover, additional steps and optimizations are required to ensure temporal consistency on the geometry mesh [Arnab et al. 2019; Bogo et al. 2016; Güler and Kokkinos 2019; Xu et al. 2018, 2019], including, in some cases, multi-view cameras to learn a geometry-aware representation of human shape, where the 3D pose is estimated through direct regression [Rhodin et al. 2018a], or the employment of networks that can learn and predict the human dynamics [Kanazawa et al. 2019; Zhang et al. 2019]. However, these works do not guarantee a consistent skeleton across different frames. Even though temporal coherence of the geometric model is encouraged, phenomena such as bone stretching may appear. In order to extract a single kinematic skeleton from such a geometric model, rigging should be applied. In contrast, our work aims to reconstruct a set of joint rotations along with a single kinematic skeleton in an end-to-end, data-driven fashion.

Unlike these methods, we utilize a single network that supports direct regression of 3D joint rotations, and at the same time ensures the temporal consistency of the skeleton, solving both problems end-to-end. More specifically, instead of fitting a skeleton to detect the 3D human pose, we integrate kinematics into the neural network, similarly to [Kanazawa et al. 2018; Pavllo et al. 2018; Villegas et al. 2018; Zhou et al. 2016], where joint angles are recovered by learning the rotational data directly. However, Zhou et al. employ loss only on joint locations, and the joint angles are estimated as a byproduct, while Kanazawa et al. recover the joint rotation angles as part of the parameters for the SMPL body model using unpaired adversarial training, with a separate discriminator for each joint. On the other hand, Pavllo et al. [2018] focus on human motion prediction by regressing the joint rotations, applying forward kinematics, and then employing a loss on joint positions. In contrast, our network learns to output temporally coherent joint angles with natural velocities using training data from available motion capture datasets, and is trained on motion sequences, as opposed to individual frames. In addition, the structure of our network enforces a single consistent

skeleton, which can be directly converted into an animation file without the need to apply IK or temporal smoothing [Arnab et al. 2019; Peng et al. 2018].

3 MOTION RECONSTRUCTION

At the crux of our approach lies a deep neural network, MotioNet, whose unique structure is inspired by the common working methodology in the animation field. MotioNet enables extraction of 3D poses with natural, data-driven, rotational information, which can be directly converted into motion file formats used in animation platforms and game engines.

In practice, our network learns to map 2D joint positions, extracted from an input video, into two separate components: (i) a single, symmetric, skeleton, represented by its bone lengths; (ii) a dynamic, skeleton-independent, sequence of: joint rotations, global root positions and foot contact labels. These two parts constitute a complete, global, description of motion, requiring no further processing, or IK, to obtain a full 3D animation.

To train our network, we leverage motion capture datasets that comprise temporal sequences of various motions performed by different subjects, where each motion is represented by a single skeleton and a temporal sequence of 3D joint rotations and global positions. During training, the 3D motion is projected onto 2D planes from arbitrary view angles, and the network is trained to recover the aforementioned components from the projected sequence. The foot contact labels are extracted from the captured motions by examining the distance of the foot joints from the ground and their velocity. During test time, the predicted labels can used to constraint the position of the joints, via IK optimization

Our main technical contribution is a new 2-branch network that reconstructs the dynamic properties of the motion – joint rotations, global root positions, foot contact labels, separately from the static ones – a single kinematic skeleton. Our network consists of various components, some of which were previously proposed by other methods, while others are new. The forward kinematics layer that we use was originally proposed by Villegas et al. [2018] in the context of motion retargeting. However, we are the first to use it for pose estimation. In addition, Kanazawa et al. [2018] already applied a discriminator to joint rotation angles, but did so in a per-frame fashion and using absolute angle values. In contrast, our discriminator judges the realism of temporal sequences of angular velocities. Lastly, Pavllo et al. [2019] have also used 1D temporal convolution in the context of 3D pose estimation. However, their network is trained to lift 2D joint positions into 3D, while ours is trained to convert joint positions to rotations.

3.1 Network Architecture

Let $\mathbf{P}_{s,q,r} \in \mathbb{R}^{T \times 3J}$ denote a temporal sequence of 3D joint positions generated by a skeleton $\mathbf{s} \in \mathbb{R}^L$ with joint rotations $\mathbf{q} \in \mathbb{R}^{T \times QJ}$ and global root positions $\mathbf{r} \in \mathbb{R}^{T \times 3}$, where L is the number of limbs, T is the temporal length of the sequence, J is the number of joints, and Q is the size of the rotations representation vector, which will be discussed later.

Our goal is to train a deep neural network to decompose sequences of projections of 3D joints, $C(\mathbf{P}_{s,q,r}; c) \in \mathbb{R}^{T \times 2J}$ (C is a perspective projection operator, which projects the 3D joint positions into 2D using camera parameters $c \in C$), into four attributes: $\tilde{\mathbf{q}} \in \mathbb{R}^{T \times QJ}$, that captures the dynamic, rotational information of the motion, $\tilde{\mathbf{s}} \in \mathbb{R}^L$, that describes the single, consistent, skeleton, $\tilde{\mathbf{r}} \in \mathbb{R}^{T \times 3}$ that estimates the global positions of the root along time, and $\tilde{\mathbf{f}} \in \{0, 1\}^{T \times 2}$ that predicts whether each of the two feet touches the ground in each frame. The first three attributes can be then combined via forward kinematics to estimate the global 3D pose sequence, $\tilde{\mathbf{P}}_{\tilde{s}, \tilde{q}, \tilde{r}} \in \mathbb{R}^{T \times 3J}$, specified by joint positions.

A high level diagram of our approach is shown in Figure 3. During training, each data sample, $C(\mathbf{P}_{s,q,r}; c) \in \mathbb{R}^{T \times 2J}$, is fed, in parallel, into two networks, E_S and E_Q. The task of E_s is to estimate the single skeleton

$$\tilde{\mathbf{s}} = E_S(C(\mathbf{P}_{s,q,r}; c)), \tag{1}$$

while E_Q aims to estimate rotations, global root positions, and foot contact labels via

$$\tilde{\mathbf{q}}, \tilde{\mathbf{r}}, \tilde{\mathbf{f}} = E_Q(C(\mathbf{P}_{s,q,r}; c))], \tag{2}$$

respectively. We train a single network to estimate all the three dynamic attributes due to the strong correlation between them (e.g., the global root position of a running human, is temporally correlated with the rotations of the limbs, and the timing of contact with the ground).

Since the rotations, global positions and contact labels are dynamic (change per frame), we design E_Q to retain temporal information. In practice, it employs one-dimensional convolution layers over the time axis, yielding a temporal set of joint rotations. In contrast, the skeleton is a static attribute, and E_S employs adaptive pooling to collapse the temporal axis, resulting in a fixed size vector of bone lengths, regardless the input dimensions, as illustrated in Figure 5.

Forward Kinematics. In order to convert $\tilde{\mathbf{s}}$, $\tilde{\mathbf{q}}$ and $\tilde{\mathbf{r}}$ into 3D joint positions, we use Forward Kinematics (FK), which refers to the process of computing the joint positions, given the joint rotations and the initial skeleton. FK is performed by successively rotating the limbs of an input skeleton tree starting from the root joint and ending in the leaf joints (see Figure 4). Since these are linear operations, we can integrate them into the network as a differential layer [Villegas et al. 2018] FK, which allows back propagation. In practice, given the estimated bone lengths, $\tilde{\mathbf{s}}$, we first reconstruct a "T-pose", $\tilde{\mathbf{s}}_{\text{init}} \in \mathbb{R}^{3J}$, by positioning the joints in 3D coordinates. This is done by placing the root at the origin, while the other joints are placed on the XY plane, based on the offsets specified in $\tilde{\mathbf{s}}$ (as demonstrated by the left most character in Figure 4). Then, at each time step t, the FK layer is applied to $\tilde{\mathbf{s}}_{\text{init}}$, using the joint rotations of the t-th frame, $\tilde{\mathbf{q}}^t \in \mathbb{R}^{4J}$ to obtain

$$\tilde{\mathbf{P}}_{\tilde{s}, \tilde{q}, \tilde{r}=0}^t = FK(\tilde{\mathbf{s}}_{\text{init}}, \tilde{\mathbf{q}}^t), \tag{3}$$

where $\tilde{\mathbf{P}}_{\tilde{s}, \tilde{q}, \tilde{r}=0}^t$ is the local 3D pose (root positioned in the origin) at frame t. Rotations are applied successively to each joint starting from the root and ending at the leaf joints, $\tilde{\mathbf{P}}_n^t = \tilde{\mathbf{P}}_{\text{parent}(n)}^t + R_n^t \tilde{\mathbf{s}}_n$, where $\tilde{\mathbf{P}}_n^t \in \mathbb{R}^3$ is the position of the n-th joint at time t, $\tilde{\mathbf{P}}_{\text{parent}(n)}^t \in \mathbb{R}^3$ is the position of its parent, $R_n \in \mathbb{R}^{3 \times 3}$ is the rotation matrix of the n-th joint, and $\tilde{\mathbf{s}}_n \in \mathbb{R}^3$ is the 3D offset of the n-th joint relative to
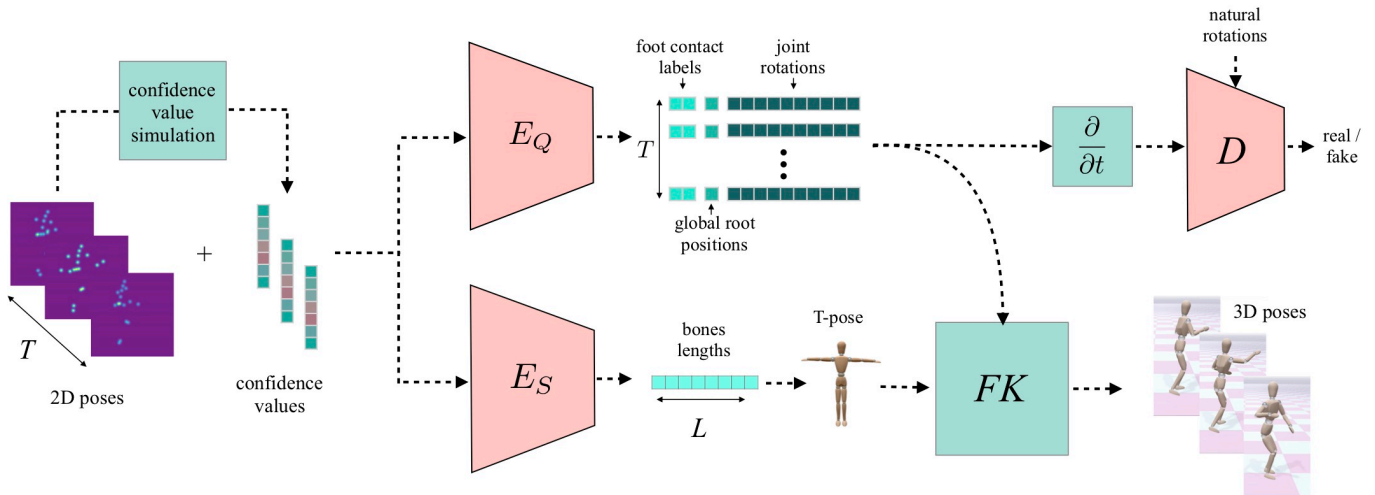
Fig. 3. Our framework receives 2D joint positions along with per-joint confidence values, which are simulated based on empirical experiments from real videos. It extracts per-frame joint rotations and global root positions along with foot contact labels, and a static (duration-independent) skeleton, using two encoders, E_Q and E_S. The extracted rotations are fed into a discriminator D that is trained to tune the temporal differences of the rotation angles to mimic the distribution of natural rotations, using adversarial training. In addition, the rotations and the static feature that is converted to a "T-pose", are fed into the forward kinematic layer, FK, that extracts 3D joint positions, which are compared to a ground truth.
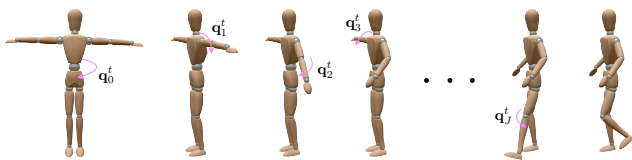

Fig. 4. Our network applies forward kinematics on a "T-pose" skeleton by successively rotating the limbs from the root to the end-effectors.

its parent in $\tilde{s}_{init}$. Finally, given the local pose of the 3D skeleton, we add the global translation to obtain the full estimated 3D pose in frame t,

$$\tilde{\mathbf{P}}^t_{\tilde{s},\tilde{q},\tilde{r}} = FK(\tilde{s}_{init}, \tilde{\mathbf{q}}^t) + \tilde{\mathbf{r}}^t. \tag{4}$$

Since FK consists of differential operators, it can be integrated into the network as a layer through which back-propagation may be applied. The FK process for a given time step is illustrated in Figure 4.

3.2 Training and Loss

To train our network, we leverage publicly available motion capture datasets described in more detail in Section 3.3. We use a combination of full self-supervised and partial self-supervised training. For the fully supervised training we use captured human motions for which the full, rotational and positional 3D joint data is available, while the partially supervised training only makes use of 3D joint positions (positional representation). In both cases, the 2D inputs are generated automatically by projecting the 3D training data, thus the paired training is essentially self-supervised.

The loss function that we use consists of four components: skeleton loss $\mathcal{L}_S$, global root position loss $\mathcal{L}_R$, joint position loss $\mathcal{L}_P$, and

rotations GAN loss $\mathcal{L}_{Q\_GAN}$. These components are described in more detail below.

Skeleton Loss. Let $\mathbf{P}_{s,q,r} \in \mathcal{P}$ denote a 3D motion sequence in our dataset $\mathcal{P}$, and $C(\mathbf{P}_{s,q,r}, c_i)$ denote its 2D projection using a camera $c_i \in C$. The skeleton loss $\mathcal{L}_S$ ensures that the encoder E_S correctly extracts the skeleton $\mathbf{s}$:

$$\mathcal{L}_S = \mathbb{E}_{\mathbf{P}_{q,s,r} \sim \mathcal{P}, c_i \sim C} \left[\|E_S(C(\mathbf{P}_{s,q,r}, c_i)) - \mathbf{s}\|^2 \right]. \tag{5}$$

When perspective projection is used, the 3D skeleton can be reconstructed only up to a scale factor. In order to avoid this scale ambiguity, we globally rescale all the 3D skeletons in the data such that their average bone length is equal to 1. At inference time, if the real spine length of the performer is known, the reconstructed 3D pose can be rescaled into a real metric space.

Joint Position Loss. This is our main loss, designed to ensure that the joints in the extracted poses are placed in the right 3D positions. Unlike naive supervised reconstruction of 3D joint positions, in this case, the positions are constrained by the geometric structure of the reconstructed skeleton, which leads to smaller, correlated, search spaces. Although the first part of the network (E_S and E_Q) is supposed to output the full, positional and rotational description of the motion, the joint position loss is essential since a direct loss over the joint rotations is not applied. In addition, since errors can be accumulated during FK from the root to the end-effectors, the FK-reconstructed joint positions, per frame, tune the joint rotations such that the reconstructed 3D joint position match the original ground truth ones:

$$\mathcal{L}_P = \mathbb{E}_{\mathbf{P}_{q,s,r} \sim \mathcal{P}, c_i \sim C} \left[\|FK(\tilde{s}_{init}, \tilde{\mathbf{q}}) - \mathbf{P}_{s,q,r=0}\|^2 \right]. \tag{6}$$

Since human observers are particularly sensitive to the motion of end effectors in animations, we split the joints into two groups and penalize the end-effectors part, $\mathcal{L}_{\mathrm{P}_{EE}}$ with a different weight ($\lambda_{P_{EE}}$).

Adversarial Rotation Loss. Applying a loss (only) over the skeleton and joint positions, trains the network to reconstruct correct positions, without imposing constraints on the rotations. Due to the inherent ambiguity that exists in IK, the resulting rotations and their velocities might be unnatural, which can lead to distortions when skinning is applied to the skeleton. Although a part of the data contains 3D rotations that were captured in motion capture systems, a direct loss applied to the rotations may not converge. Due to the fact that the T-poses for different samples in the dataset are not aligned, namely, two similar poses might be represented by different rotations, thus, a direct loss on the rotations can not be applied, unless, the entire set is retargeted to share the T-pose, which is a time consuming operation. Due to the potential difference between the absolute values of rotations that represent the same pose, our network is trained to output rotations with natural velocities distribution using adversarial training. The idea is to focus on the temporal differences of joint rotations rather than their absolute values. For example, if a specific joint is constrained to move only in a specific direction we want to follow that rule, in a data-driven manner. In order to incorporate the adversarial loss we add J discriminators, D_j, where each is aimed to distinguish between temporal differences of the output rotations and real rotations from the dataset for a specific joint,

$$
\begin{aligned}
\mathcal{L}_{\mathrm{Q\_GAN}_j} = \quad & \mathbb{E}_{q \sim Q} \left[\| D_j(\Delta_t q_j) \|^2 \right] \\
+ \quad & \mathbb{E}_{\mathbf{P}_{q,s,r} \sim \mathcal{P}, c_i \sim C} \left[\| 1 - D_j(\Delta_t E_Q(C(\mathbf{P}_s, q, r, c_i)))_{q_j} \|^2 \right],
\end{aligned}
\tag{7}
$$

where $E_Q(\cdot)_{q_j}$ denotes the rotations of the jth joint of the output of E_Q, Q represents the distribution of natural joint angles that are taken from the dataset, and Δ_t stands for temporal differences. The full architecture of each components can be found in the Appendix A.

Note that due to the averaging behavior of the temporal convolutions and the temporal continuity of the input sequence, the estimated joint rotations are encouraged to be smooth along the temporal axis, resulting in smooth animations.

Reference T-pose. Since our rotations are guided by velocities, their absolute values might be unnatural and erroneous. These errors can be observed when skinning is applied to the underlying skeleton. In order to let the network learn a correct reference point, in each epoch, we train the network with a few T-poses (T-pose per character), and require that their reconstructed rotations will be zero using a direct loss over the output rotations.

Global Root Position Loss. In order to avoid the learning of unnecessary information in the reconstruction of the global root position in frame t, $\mathbf{r}^t = (X_r^t, Y_r^t, Z_r^t)$, we estimate only the depth parameter Z_r^t and then derive the global root position by

$$
(X_r^t, Y_r^t, Z_r^t) = \left(\frac{Z_r^t}{f} x_r^t, \frac{Z_r^t}{f} y_r^t, Z_r^t \right),
\tag{8}
$$

where (x_r, y_r) is the root position in the projected 2D input sequence, and $f = 1$ is the assumed focal length of the camera model. Then

the loss is given by

$$
\mathcal{L}_{\mathrm{R}} = \mathbb{E}_{\mathbf{P}_{q,s,r} \sim \mathcal{P}, c_i \sim C} \left[\| E_Q(C(\mathbf{P}_s, q, r, c_i))_r - Z_r \|^2 \right],
\tag{9}
$$

where $E_Q(\cdot)_r$ denotes the root position part of the output of E_Q.

Foot Contact Loss. In order to handle foot sliding artifacts we supervisedly train the network to output $\tilde{\mathbf{f}}$ that predicts whether each foot is in contact with the ground in each frame via

$$
\mathcal{L}_{\mathrm{F}} = \mathbb{E}_{\mathbf{P}_{q,s,r} \sim \mathcal{P}, c_i \sim C} \left[\| E_Q(C(\mathbf{P}_s, q, r, c_i))_f - \mathbf{f} \|^2 \right],
\tag{10}
$$

where $E_Q(\cdot)_f$ denotes the foot contact label part, $\tilde{\mathbf{f}} \in \{0, 1\}^{T \times 2}$, of the output of E_Q. Based on that binary GT vector, $\mathbf{f}$, another loss, that encourages the velocity of foot positions to be zero during contact frames, is added

$$
\mathcal{L}_{\mathrm{FC}} = \mathbb{E}_{\mathbf{P}_{q,s,r} \sim \mathcal{P}, c_i \sim C} \left[\| \mathbf{f}_i \sum_j \Delta_t FK(\tilde{\mathbf{s}}_{\mathrm{init}}, \tilde{\mathbf{q}})_{f_i} \|^2 \right],
\tag{11}
$$

where $FK(\cdot, \cdot)_{f_i} \in \mathbb{R}^{T \times 3}$ and $\mathbf{f}_i$ denote the positions and the contact labels of one of the feet joints ($i \in left, right$), and $\sum_j$ sums the components along the coordinate axis. The contribution of this component is evaluated in the ablation study in Section 5.2.

Summing the four terms, we obtain our total loss:

$$
\mathcal{L}_{\mathrm{tot}} = \mathcal{L}_{\mathrm{P}} + \lambda_{\mathrm{S}} \mathcal{L}_{\mathrm{S}} + \lambda_{\mathrm{Q}} \sum_{j \in J} \mathcal{L}_{\mathrm{Q\_GAN}_j} + \lambda_{\mathrm{R}} \mathcal{L}_{\mathrm{R}} + \lambda_{\mathrm{F}} \mathcal{L}_{\mathrm{P}_F} + \lambda_{\mathrm{FC}} \mathcal{L}_{\mathrm{P}_{FC}},
\tag{12}
$$

where in all of our experiments $\lambda_{\mathrm{S}} = 0.1$, $\lambda_{\mathrm{Q}} = 1$, $\lambda_{\mathrm{R}} = 1.3$, $\lambda_{\mathrm{F}} = 0.5$, $\lambda_{\mathrm{FC}} = 0.5$.

3.3 Motion Dataset

Our training data is constructed from the CMU motion capture dataset [CMU 2019], and from the Human3.6M human pose dataset [Ionescu et al. 2014]. The CMU dataset consists of 2605 captured motions performed by 144 subjects. The motions include elementary actions (walking, jumping, kicking, etc.), and various dancing moves (samba, salsa, etc.). The Human3.6M dataset consists of over 3.6 million 3D human poses with corresponding images, viewed from four different angles using synchronized cameras. The motions are performed by 11 actors and include 17 everyday scenarios, such as eating, smoking, talking on the phone, walking, etc. The actors in both datasets feature a variety of body shapes and skeleton proportions.

In order to train the network to extract foot contact labels in a supervised fashion, we must first extract "ground truth" labels from the dataset. Specifically, we consider a foot to be in contact if its height above the ground is below 20mm, and its average velocity magnitude in the surrounding 5-frame window is below 1mm/frame. The height of the ground is approximated by the average of the lowest 20-percentile of all the foot heights in the sequence.

The actual input data samples fed into the network are generated by projecting the sequences of 3D joint positions (represented in camera coordinates space) onto different camera view angles, using perspective projection. As a result, we obtain a rich dataset with ground truth, which demonstrates how skeletons of different characters, performing various motions, appear from different views when projected into 2D.
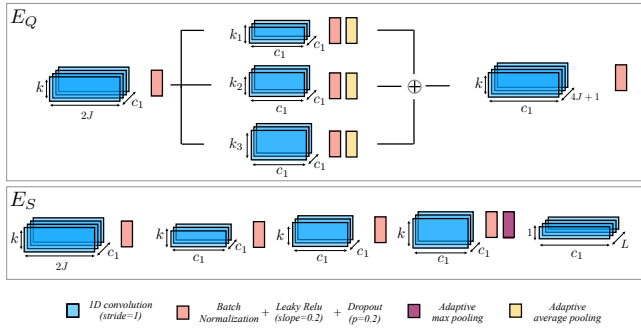
Fig. 5. Our network contains two encoders, E_Q which generates a temporal set of joint rotations, global positions and foot contact labels using parallel convolutions, and E_S which outputs a static attribute that represents the skeleton, using an adaptive pooling layer that collapses the temporal axis.

In test time, when videos in the wild are used, we use the method of Cao et al. [2018] to extract 16 joints (root, neck, thorax, head, shoulders, hips, knees, feet, elbows, and hands). These joints exist in the aforementioned datasets as well. In addition, a spine joint which exists only in the 3D datasets is artificially added (calculated as the 2D spatial average between the root and the neck joint).

Preprocessing. In order to normalize the 2D input data we apply two operations: (i) Subtract the root position from every joint location (in all the frames), (ii) Subtracting the mean joint position and divide it by the standard deviation, while the mean and standard deviation are empirically calculated per joint over the entire dataset. Since global information is discarded from the normalized local representation, we append to it the global 2D velocity (per-frame).

3.4 Implementation Details

In practice, our implementation consists of two encoders, E_Q and E_S, a forward kinematics layer FK, and a discriminator D. The layers and the dimensions of the two encoders are shown in Figure 5, while the FK layer is described by Villegas et al. [2018]. For a detailed description of the parameters of each layer, please refer to the appendix.

In the E_Q encoder there are five 1D convolution layers. The first layer expands the number of channels, and the result is then fed in parallel into three convolution layers with kernels of three different sizes. This is inspired by the inception module from GoogLeNet. The three results are then summed together after adaptive pooling and fed into a final layer that generates the temporal sequence of quaternions and root depths.

The E_S encoder uses a simple sequence of 1D convolution layers. Again, the first layer expands the number of channels, and the last layer outputs a vector of bone length, which implicitly enforces symmetry, since each pair of symmetric limbs is encoded by the same coordinate.

The discriminator D also uses a sequence of 1D convolution layers. There is a per-joint convolution layer which is summarized into a fully connected layer that dictates whether the rotational velocity is real or fake.

4 SUPPORTING VIDEOS IN THE WILD

During training, the input to our network is produced by projecting mostly clean motion captured data from 3D to a 2D image plane. However, at test time, the network should be able to cope with 2D joint positions extracted from videos in the wild, using the OpenPose library [Cao et al. 2018]. In order to overcome the gap between the distribution of the training inputs and the test-time ones, we employ confidence values and augmentation.

Confidence Values. During test time, when 2D joint positions are extracted from videos in-the-wild, there might positional errors or even missing joints due to occlusion. In order to simulate such input characteristics during the training, we leverage the output structure that state-of-the-art 2D pose estimators provide and attach for each joint a value $c_n \in [0, 1]$, which specifies the confidence in the estimation of that joint's 2D position. For a completely missing joint the confidence c_n is set to 0.

Thus, in practice, we concatenate another vector $\mathbf{c} \in \mathbb{R}^{J \times T}$ to our input, both during training and at test time. At test time the values of $\mathbf{c}$ are extracted from the output of Cao et al. [2018], while during training we randomly draw the values, based on an empirically determined distribution. The distribution was determined using the confidence values that were extracted from a diverse set of videos in-the-wild taken from [Johnson and Everingham 2010]. We calculate the histogram of each joint separately, and model each distribution as a sum of two parts: a delta distribution around zero (with probability of δ_j) and a Gaussian distribution (μ_j, σ_j) clipped at 1 (probability of $1 - \delta_j$). Figure 6 plots the normalized histograms of the confidence values for 4 different joints and the per-joint parameters $(\delta_j, \mu_j, \sigma_j)$ are listed in the appendix. It may be seen that joints that are closer to the end-effectors tend to be occluded more frequently.

Once a confidence value $c_n > 0$ is drawn, we compute a spatial displacement, which is applied as a noise to the 2D coordinate. The direction of the displacement is drawn uniformly from $[0°, 360°]$, while the magnitude is drawn from $[0, \beta(1 - c_n)]$ for $\beta = 6$. For $c_n = 0$ we set the value of the coordinate to $(0, 0)$. Thus, similarly to denoising autoencoders, our network is trained to be robust to inaccurately estimated, or even missing joint locations.

Augmentation. To enrich the observed samples, we apply data augmentation in two different ways:

(1) Clip length: since our model is fully convolutional, input motion clips with varying lengths can be forwarded through the same network. However, due to the use of batches a fixed temporal length can improve training time. Thus, we randomly pick the temporal length (T) of the samples in the batch, per iteration, to be an integer in the range of 60 to 200. This augmentation step enhances the disentanglement of the static parameters and the temporal length of the sequences.

(2) Camera augmentation: since we work in the 3D camera coordinate space, our projections are augmented by modifying the depth of the 3D character (global translation along the z axis) and its orientation (global rotation of the root). This operation is important since it trains the network to map similar poses in different 2D scales into the same (local) 3D
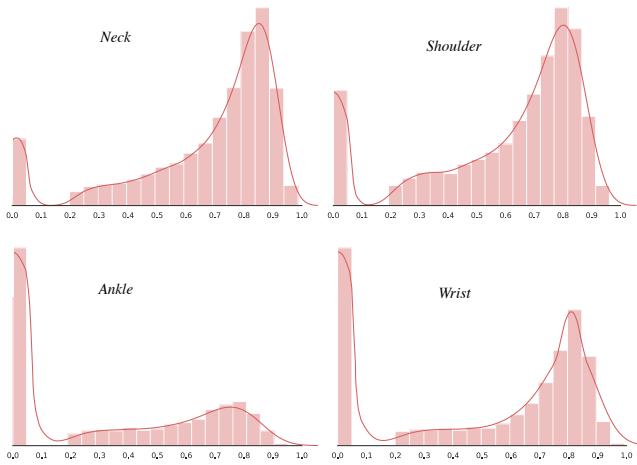
Fig. 6. Modeling the distribution of joint confidence values (continuous red line) using empirical distribution of confidence values (bins), extracted from videos in the wild.
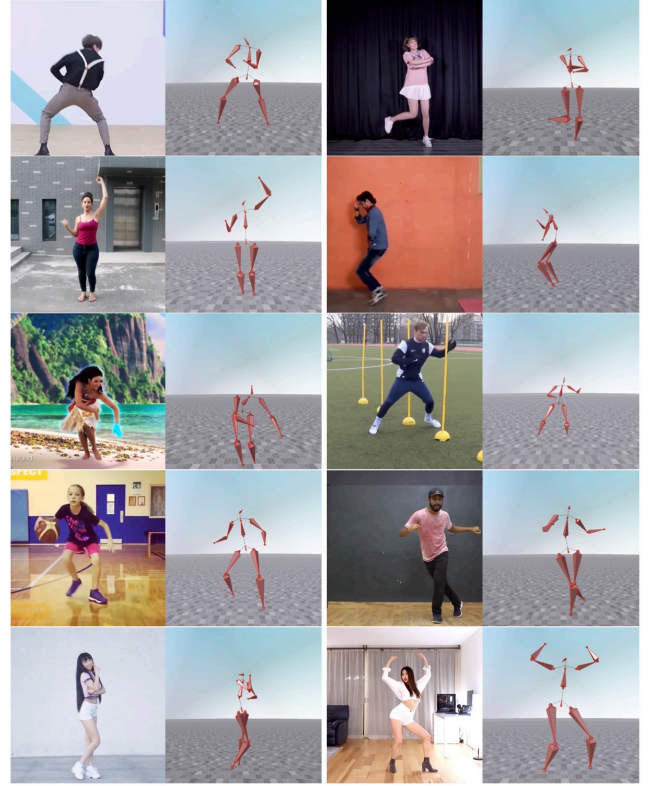

Fig. 7. Examples of motions reconstructed by MotioNet from videos in-the-wild. MotioNet can reconstruct a variety of human motions, while being robust to partial occlusions and tolerant to different body types. Frames from the video are on the left, and the reconstructed pose is on the right.

parameters (rotations and bones length), thus extending the distribution of the inputs of the training data.

(3) Flipping: we left-right flip the 3D joint positions (camera coordinate space) of the samples, before the projections, to obtain augmented poses with $\tilde{P}_j^r = \left(-(P_j^l)_x, (P_j^l)_y, (P_j^l)_z\right)$, $\tilde{P}^l = \left(-(P_j^r)_x, (P_j^r)_y, (P_j^r)_z\right)$, were P_j^r and P_j^l are the left and right positions of a symmetric joint j (e.g, left and right shoulder). For single joints without symmetry (root, neck, etc.) we use $\tilde{P}_j = \left(-(P_j)_x, (P_j)_y, (P_j)_z\right)$.

5 RESULTS AND EVALUATION

In this section, we show 3D human motion reconstruction results obtained by MotioNet, conduct experiments and ablation studies to analyze the performance of various components in our framework and present comparisons to state-of-the-art 3D human motion estimation techniques. Figure 7 shows a gallery of results which demonstrate the ability of MotioNet to extract a variety of human motions from videos in the wild, while being robust to partial occlusions and tolerant to different body types. In order to produce the animations, we convert the tensors produced by our network into BVH files, and use the Blender software to read these files and render the videos, without any further post processing. The quality of the resulting animations may be examined in the supplementary video.

5.1 Comparison to Other Methods

We first compare the accuracy of the reconstruction of joint positions by our approach to a variety of existing methods.

Quantitative comparison. A quantitative comparison is reported in Table 1, which reports the mean per joint position error (MPJPE):

$$E(\tilde{\mathbf{P}}, \mathbf{P}) = \frac{1}{J} \sum_{j=1}^{J} \|\tilde{\mathbf{P}}_j - \mathbf{P}_j\|. \tag{13}$$

We evaluated our method on two datasets. The first one is Human3.6M dataset [Ionescu et al. 2014] which adopts a 17-joint skeleton representation. For this dataset, the network was trained on five subjects (S1, S5, S6, S7, S8), and tested on two (S9 and S11), using a single model that was trained for all actions. The second dataset is HumanEva-I [Sigal et al. 2009] which adopts a 15-joint skeleton representation, including three different subjects. We evaluated our system on two actions (Walk, Jog) by training a single model. Note that similarly to other approaches, the evaluation with this dataset is computed after performing a rigid alignment with the ground truth including scale, rotation, and translation (a.k.a P-MPJPE).

In order to apply a fair comparison we did not simulate the confidence map, and used 2D detected key-points during both training and test time. We followed Pavllo et al. [2019] and used the cascaded pyramid network (CPN) [Chen et al. 2018b] to extract 2d poses. The bounding box required by CPN is extracted by Mask-RCNN [He

et al. 2017] and the 2d detector is fine-tuned by 2d ground truth samples from Human3.6M.

As may be seen from the results in Table 1 and Table 2, while our method does not achieve the highest accuracy for joint positions, it is usually among the top 2–3 methods. This is despite the fact that our method aims at a slightly different task. Furthermore, note that enforcing a single underlying skeleton may introduce some errors into the reconstructed joint positions due to inaccuracies (even small ones) in the estimated bone lengths. In order to measure the effectiveness of the the rotations branch and eliminate the effect of errors in the skeleton reconstruction, we conduct an experiment that uses the ground truth (GT) skeleton instead of the reconstructed one. In this experiment only E_Q was trained, while FK received the GT skeleton. The result reported in Table 1 shows that if the exact skeleton is known, the accuracy of poses can be even higher.

In order to demonstrate the effectiveness of our dynamic encoder E_Q, we replaced its structure by the dilated 1-D convolution architecture proposed by Pavllo et al. [2019], originally designed to output positions. The results which demonstrate that our architecture fits better to the prediction of rotations are also depicted in Table 1.

Qualitative comparison. We next conducted qualitative comparison with methods that aim to reconstruct joint rotations. A qualitative quantitative comparison is not enabled in that case since these different methods use a different number of joints and the rotation values might be ambiguous. The comparison is done against 3 methods: VNECT [Mehta et al. 2017b], HMR [Kanazawa et al. 2018] and Pavllo et al. [2019], and may be seen in the supplementary video.

To compare the reconstructed motions qualitatively to Pavllo et al. [2019], which output joint positions, the positions have to be converted to rotations. This conversion task is challenging due to ambiguities that are caused by the absence of the roll axis, temporally inconsistent skeleton size, and bone length violations that amplify the reconstruction errors and lead to abnormalities in the resulting animation.

In our comparison, we employ the rudimentary method of Dong et al. [2017], which iteratively computes the local joint angles using intrinsic rotations.

Mehta et al. [2017b] use Inverse kinematics (IK) to calculates joint rotations and fit a single skeleton to given joint positions. Compared to position-to-rotation conversion using intrinsic rotations [Dong et al. 2017], IK yields more stable rotations suitable for animation, however, it limits the degrees of freedom to regulate the joint rotations, restricting the nature of motion.

Kanazawa et al. [2018] learn joint rotations in an adversarial manner in a per frame fashion. While their approach is also data driven, with no temporally consistent rotations, an animation can not be extracted directly from their per frame output. In addition, in order to extract an animation, an average skeleton should be calculated.

The full comparison can be found in the supplementary video, and some extracted frames are depicted in Figure 8. It can be seen that the combination of Pavllo et al. [2019] and Dong et al. [2017] exhibits abnormalities in the rotations resulting from the converted rotations, while the animations extracted by MotioNet do not suffer

from the same issues. In addition, while Kanazawa et al. [2018] produce inconsistent rotations, especially next to the end-effectors (as evidenced by shaking limbs), our reconstruction yields plausible stable global motions. Due to their per-frame adversarial loss, the rotations reconstructed by their method are not temporally coherent. In order to extract a stable animation form their method, temporal smoothing and IK optimization is required, as has been done in Peng et al. [2018].

Overall, these experiments show that MotioNet can recover joint positions comparably to state-of-the-art pose estimation methods, while generating more natural, temporally consistent joint rotations and plausible animations, compared to other existing methods.

Bone Lengths Estimation. In this experiment we evaluate the accuracy of the bone lengths estimation and compare the results with other methods, including VNECT [Mehta et al. 2017b], HMR [Kanazawa et al. 2018], Martinez et al. [2017], and Pavllo et al. [2019]. We tested the methods using Human3.6M and measured the results against the ground truth skeleton, where each bone length was calculated by the Euclidean distance between the two relevant joints. In order to avoid the scaling bias, a normalization is applied to all the skeletons during training, while at test time we de-normalize the output skeletons using the saved scaling factor and calculate the error against the original ground truth. The final error values, which are calculated as the average of all the bone errors, are plotted in Figure 9. It can be seen that our network E_S outperforms previous methods in the task of bone lengths estimation.

Global Root Positions Reconstruction. In this experiment we evaluate the reconstruction of the global root positions and compare it to VNECT [Mehta et al. 2017b]. The evaluation was done using the test set of Human3.6M. Since the scaling factor of the reconstructed root position of VNECT is unknown, we avoid a quantitative comparison in that case. In order to qualitatively evaluate the reconstruction we visualize the positions of the root coordinate along time for some motions, where the marker becomes darker when time progresses, as shown in Figure 10. The root path is visualized from another point of view (different than the camera's perspective) to demonstrate that our depth estimation is reliable (which can not be observed from the camera perspective). More qualitative results can be found in supplementary video.

Online Motion Reconstruction. In this experiment we demonstrate the potential of our framework to be a core part of an online motion reconstruction system. In order to conduct such an experiment, we replaced our convolutions with casual convolutions, namely, convolution kernels that consider only past frames, while the kernels keep their original dimensions. The results in Table 1 show that such a setup still enables to achieve good reconstruction. However, note that the runtime during inference, in practice, also depends on other factors, such as the 2D key-points extractor.

Rotation Representation. 3D rotations may be represented with Euler angles (3D) or unit quaternions (4D). Recently, Zhou et al. [2019] proposed a new 6D representation for rotations, which might be helpful for training deep neural networks due to the inherent discontinuities that exist in other representations. Since our framework

	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	avg.(mm)
Mehta et al. [2017b]	62.6	78.1	63.4	72.5	88.3	93.8	63.1	74.8	106.6	138.7	78.8	73.9	82.0	55.8	59.6	80.5
Pavlakos et al. [2017]	67.4	71.9	66.7	69.1	72.0	77.0	65.0	68.3	83.7	96.5	71.7	65.8	74.9	59.1	63.2	71.9
Martinez et al. [2017]	51.8	56.2	58.1	59.0	69.5	78.4	55.2	58.1	74.0	94.6	62.3	59.1	65.1	49.5	52.4	62.9
Yoshiyasu et al. [2018]	63.3	71.6	61.4	70.4	69.9	83.2	63.0	68.8	76.8	98.9	68.2	67.5	73.7	57.7	57.1	69.9
Fang et al. [2018]	50.1	54.3	57.0	57.1	66.6	73.3	53.4	55.7	72.8	88.6	60.3	57.7	62.7	47.5	50.6	60.4
Yang et al. [2018](+)	51.5	58.9	50.4	57.0	62.1	65.4	49.8	52.7	69.2	85.2	57.4	58.4	43.6	60.1	47.7	58.6
Hossain et al. [2018](†)	48.4	50.7	57.2	55.2	63.1	72.6	53.0	51.7	66.1	80.9	59.0	57.3	62.4	46.6	49.6	58.3
Pavlakos et al. [2018a](+)	48.5	54.4	54.4	52.0	59.4	65.3	49.9	52.9	65.8	71.1	56.6	52.9	60.9	44.7	47.8	56.2
Lee et al. [2018a](†)	43.8	51.7	48.8	53.1	52.2	74.9	52.7	44.6	56.9	74.3	56.7	66.4	47.5	68.4	45.6	55.8
Habibie et al. [2019]	46.1	51.3	46.8	51.0	55.9	43.9	48.8	65.8	81.6	52.2	59.7	51.1	40.8	54.8	45.2	53.4
Chen et al. [2019]	43.8	48.6	49.1	49.8	57.6	61.5	45.9	48.3	62.0	73.4	54.8	50.6	56.0	43.4	45.5	52.7
Mehta et al. [2019](+)	50.2	61.9	58.3	58.2	68.8	74.6	54.1	61.5	76.8	91.7	63.4	58.5	48.3	65.3	53.2	63.6
Pavllo et al. [2019](†)	45.2	46.7	43.3	45.6	48.1	55.1	44.6	44.3	57.3	65.8	47.1	44.0	49.0	32.8	33.9	46.8
Single 1-D conv, gt-skeleton(†)	55.89	61.67	58.88	62.43	62.10	61.36	60.57	64.00	72.74	63.38	75.27	63.51	58.87	67.69	58.87	63.15
Ours, gt-skeleton(†)	43.61	49.39	46.60	50.15	49.82	49.08	48.29	51.72	60.46	51.10	62.99	51.23	46.59	55.41	46.59	50.87
Casual conv(real-time)(†)	51.85	57.63	54.84	58.39	58.06	57.32	56.53	59.96	68.70	59.34	71.23	59.47	54.83	63.65	54.83	59.11
Ours(†)	47.33	53.11	50.32	53.87	53.54	52.80	52.01	55.44	64.18	54.82	66.71	54.95	50.31	59.13	50.31	54.59

Table 1. Quantitative comparison to other joint position reconstruction methods using MPJPE on Human3.6M dataset. Legend: (†) multi-frames as an input, (+) extra training data (Yang et al. [2018] use 2D annotations from the MPII dataset. Pavlakos et al. [2018a] use additional data from the Leeds Sports Pose (LSP) dataset).
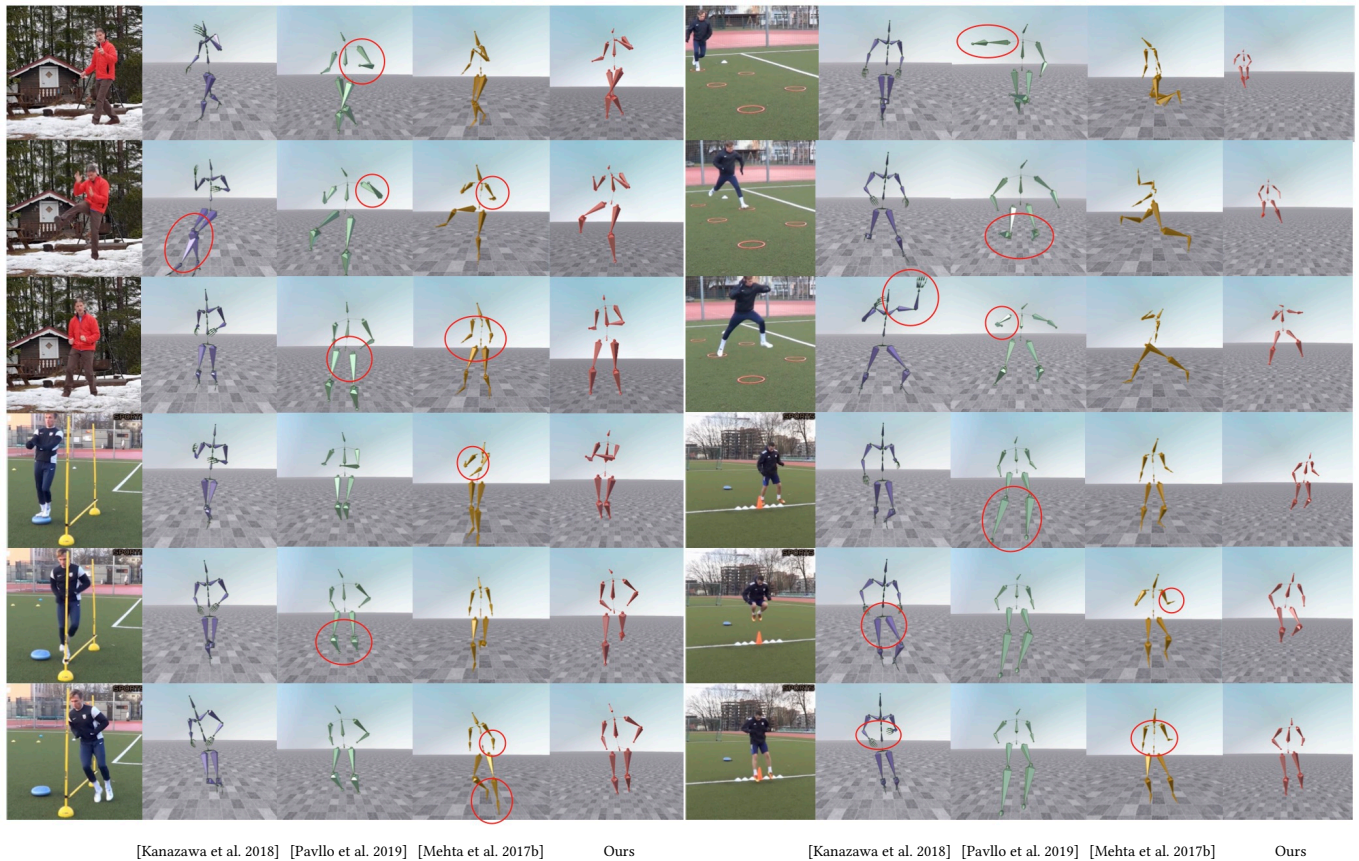

[Kanazawa et al. 2018] [Pavllo et al. 2019] [Mehta et al. 2017b] Ours [Kanazawa et al. 2018] [Pavllo et al. 2019] [Mehta et al. 2017b] Ours

Fig. 8. Our results on videos in the wild, compared to three other methods: (i) Kanazawa et al. [2018] (ii) Mehta et al. [2017b] (iii) Pavllo et al. [2019] + Dong et al. [2017]. Please see the supplementary video, where it may be seen that our method recovers plausible global motions with smooth joint rotations.

is not limited to a specific representation, we conducted an experiment that compares between these three representations. We found that the 6D representation of Zhou et al. [2019] achieves the lowest quantitative error (52.54), which is comparable to the error that is

Methods	Walk			Jog		
	S1	S2	S3	S1	S2	S3
Pavlakos et al. [2017]	22.3	15.9	**22.7**	28.9	21.9	23.8
Pavlakos et al. [2018a] (+)	18.8	12.7	29.2	23.5	15.4	14.5
Lee et al. [2018a]	18.6	19.9	30.5	25.7	16.8	17.7
Pavllo et al. [2019]	**13.9**	**10.2**	46.6	20.9	**13.1**	**13.8**
Ours	15.3	11.7	33.7	**19.8**	14.5	16.1

Table 2. Quantitative comparison to other joint position reconstruction methods using P-MPJPE on HumanEva-I. (+) extra training data.
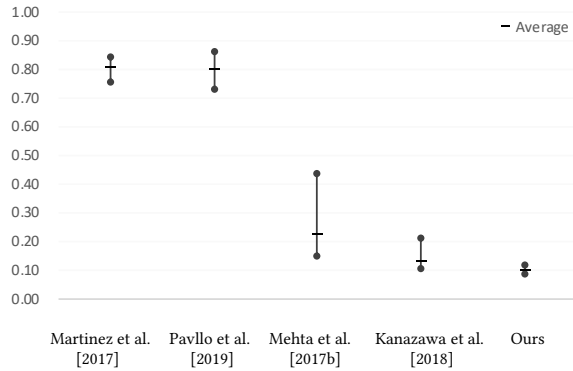

Fig. 9. Bone lengths estimation. The maximum, minimum and average bone length errors calculated for Human3.6M test subjects.
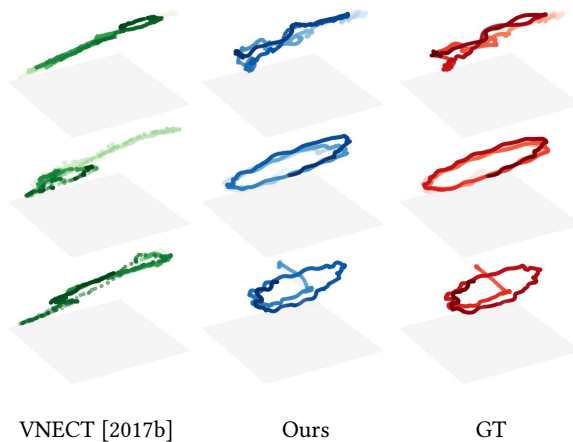

Fig. 10. Comparison of global root position reconstruction accuracy. The estimated (VNECT [Mehta et al. 2017b] - left column, ours - middle column) and the GT positions (right column), (X_r^t, Y_r^t, Z_r^t), were extracted form 3 motions in the test data (top to bottom): directions, walking and eating.

achieved by the unit quaternions (52.69), but much smaller than Euler angles (116.33). However, by qualitatively evaluating the results shown in the supplementary video, it can be seen that in practice, the results with the 6D representation, contain self-rotations and minor global shaking (note that the visualization is done without

the global root position, which is irrelevant to the comparison). In contrast, in the quaternion results this phenomenon is mitigated. For the given structure of the network, it seems that the network is more challenged in the production of stable results when the output matrix is larger. Thus, we pick unit quaternions as our representation through all the experiments.

5.2 Ablation Study

In this section we evaluate and demonstrate the effectiveness of different components and losses in our framework.

Simulated Confidence Map and Augmentation. In this experiment we compare between our results in two different setups: with and without the simulated confidence map. In order to measure the results, we used the Human3.6M dataset, which additionally to the motion files contains video frames of the motion captured actors. The frames enable to simulate the full pipeline that consists of 2D position extraction (instead of 2D projection), and to evaluate the results against ground truth. We used 239 raw videos from the test data, extracted the 2D joint positions using [Cao et al. 2018], evaluated the 3D output against the GT and reported the results in Table 3. It can be seen that although the confidence map consideration degrades the results over the clean data, it improves the results of the 2D joints extracted from videos. A qualitative comparison can be found in the supplementary video. It can be observed that without the simulated confidence map, training over clean projected data, leads to minor shakes, which are caused by the noisy input extracted from the video frames and don't appear in the training data. In addition, we perform an experiment without any data augmentation and one without the T-pose loss, and report the results in Table 3, which demonstrate the contribution of each step.

Rotational Velocities Discriminator. Here we evaluate the quality of the motion reconstruction without the $\mathcal{L}_{Q\_GAN}$ loss term, using the CMU test dataset which contains joint rotations for every sample. A qualitative comparison can be found in the supplementary video. Although the rotations are temporally smooth due to the existence of temporal convolution filters, it can be seen that the absence of the discriminator in our framework might yield unnatural self rotations (especially around the roll axis) and inconsistent rotation velocities. Figure 11 demonstrates the distribution of the temporal differences of the learned rotations in three different setups: without a discrimina tor, with a discriminator applied to rotations represented in unit quaternions and a discriminator applied to rotations represented in Euler angles. In all of the last experiments, the network outputs unit quaternions, while appropriate conversions are performed for the Euler angles experiment. The diagram depicts the quantity of the magnitude of the rotational velocity, per axis, in Euler angles representation. It can be seen that the distribution of the velocities become closer to the distribution of the original data with the attendance of the discriminator, for both representations, even though no direct loss is applied to the output rotations. In addition, Table 3 shows that the quaternion-based discriminator quantitatively performed better than the Euler angle experiments, and this is our chosen representation for the rest of our experiments.

Method	Error (GT)	Error (CPN FT)
Baseline	52.69	54.59
Baseline with augmentation	52.32	54.41
Baseline with discriminator (quaternions)	69.78	73.48
Baseline with discriminator (Euler)	70.23	74.14
Baseline without T-pose loss	55.08	57.81

Table 3. Ablation study. Various comparisons to demonstrate the effectiveness of different components and losses in our framework against the baseline. The second column depicts the error obtained by clean 2D inputs (projections of the 3D data), while the values in the third column obtained form inputs that were extracted from real video frames using fine-tuned CPN detector.

Fig. 12. Foot contact label prediction is correlated with foot positional error. Top: foot positional error, bottom: foot contact label prediction accuracy (0 - correct, 1 - incorrect).
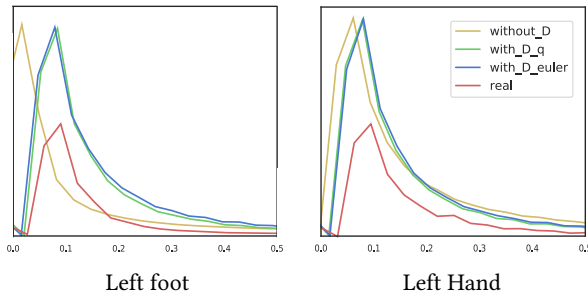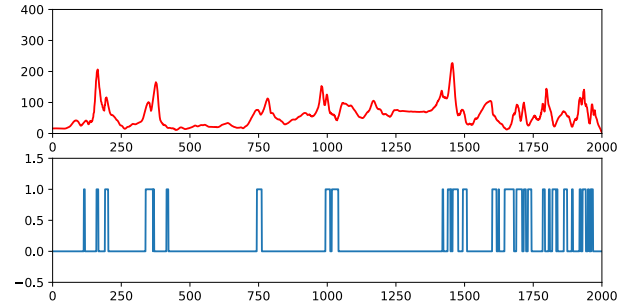

Fig. 11. Distribution of the magnitude of the joint rotational velocity of the left foot (left) and left hand (right) in our CMU test data: GT (red), outputs without a discriminator (yellow), outputs with a discriminator applied to rotations represented in unit quaternions (green) and outputs with a discriminator applied to rotations represented in Euler angle (blue).

Foot Contact Loss. We performed an experiment where the network doesn't consider the foot contact prediction and foot contact loss and show its results in the supplementary video. It can be seen that when no explicit attention is given to foot contact, perceptually disturbing foot skating can be easily observed. In particular in our case, where FK is applied in a successive order from the root to the end-effectors, the accumulated error makes the unstable feet even more noticeable. The experiment results in a foot contact error of 64.01 vs. error of 52.34 with the foot contact loss, while the error is calculated by the Euclidean distance between the output and the GT foot positions only in frames where there is a contact with the ground and averaged along time. Qualitative evaluation in the supplementary video demonstrates that the prediction helps to mitigate foot sliding artifacts. In addition, as visually demonstrated in Figure 12 (for some arbitrary slot of motion from our test data), our experiments show that there is a high correlation between the accuracy of the predicted foot label (bottom) to the error of the foot position (top). Despite the fact that the network doesn't make use of the predicted foot contact labels, high contact prediction accuracy yields better accuracy for feet positions.

6 CONCLUSION, LIMITATIONS, AND FUTURE WORK

We introduce a holistic approach that learns to extract a complete 3D human motion from monocular video, instead of inferring a sequence of human poses. Using standard skinning and post-processing, similar to the ones applied to data that was captured in a controlled environment, we can achieve even higher quality animations that can be used in various applications and game engines, as can be seen at the end of our supplementary video. Specifically, while previous works perform pose estimation, reconstructing only 3D joint positions, our deep neural network, MotioNet, is trained to recover a *single* skeleton with a sequence of 3D joint *rotations*, and, consequently, the joint positions, as well. In general, joint positions do not fully capture human motions. Human poses defined by joint positions alone may not satisfy the fixed bone length constraint. Even with the use of IK, ambiguities and temporal discontinuities can still occur. In contrast, a fixed skeleton with a temporal sequence of designated joint rotations completely characterizes the motion; it is a native representation for animation synthesis, as it often composes the subspace of the character rig space.

A key advantage of our approach is that it does not require IK: both joint positions and joint rotations are learned from training data of real human motions, rather than solved for. This leads to a more natural human motion reconstruction, via the FK layer of MotioNet. Furthermore, by inferring a single skeleton, MotioNet is guaranteed to keep the bone lengths fixed along the entire motion sequence. This provides a prior that facilitates motion reconstruction in the presence of significant occlusion in the input video. Another mechanism we employ to address the occlusion challenge is to inject random noise to the training input sequences and augment them with confidence values whose distribution mimics that of confidence values extracted by 2D pose estimation methods from videos in the wild. The network is thus trained to cope with noise and occlusions, akin to denoising autoencoders. Finally, since our system is trained in the space of motions, the inherent smoothness of human motions is learned from the data, achieving temporal coherence naturally.

By design, MotioNet only tracks the motion of a single human character. The handling of multi-character motions, especially when the characters undergo close interactions, is clearly challenging, and we leave it for future work. Also, our current method may not accurately recover the global positioning of the skeletons, e.g., when there are significant camera movements. Another limitation is that our network does not account for any physical constraints of the human character or of the interactions between the character and

the environment — the motion reconstruction is purely based on analyzing and learning from visual cues, which could be billed as an advantage as well. As a result, for example, there is no notion of contact constraints in the system, thus the resulting motion may suffer from foot sliding. This could potentially be solved by predicting the foot contact conditions and using a loss function term that evaluates the positional constraints as done by Lee et al. [2018b].

In other possible future works, we are interested in learning the subspace of human motion from the videos, so that animators can easily produce physically-plausible animations with little effort. We are also interested in learning the differences between motion styles from a large amount of video data that is currently publicly available. A dataset with motion style annotations will be useful for creating a generative model, where novel motion styles can be sampled.

ACKNOWLEDGMENTS

REFERENCES

Ijaz Akhter and Michael J. Black. 2015. Pose-Conditioned Joint Angle Limits for 3D Human Pose Reconstruction. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '15)*. IEEE Computer Society, Washington, DC, USA.

Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. 2018. Video Based Reconstruction of 3D People Models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '18)*. IEEE Computer Society, Washington, DC, USA.

Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. 2005. SCAPE: Shape Completion and Animation of People. *ACM Trans. Graph.* 24, 3 (July 2005), 408–416. https://doi.org/10.1145/1073204.1073207

Anurag Arnab, Carl Doersch, and Andrew Zisserman. 2019. Exploiting temporal context for 3D human pose estimation in the wild. In *Proceeding of the IEEE/CVF Conference in Computer Vision and Pattern Recognition (CVPR'19)*. 3395–3404.

Andreas Baak, Meinard Müller, Gaurav Bharaj, Hans-Peter Seidel, and Christian Theobalt. 2011. A data-driven approach for real-time full body pose reconstruction from a depth camera. In *Proceedings of the International Conference on Computer Vision (ICCV '11)*. IEEE Computer Society, 1092–1099. https://doi.org/10.1109/ICCV.2011.6126356

Didier Bieler, Semih Günel, Pascal Fua, and Helge Rhodin. 2019. Gravity as a Reference for Estimating a Person's Height from Video. arXiv:cs.CV/1909.02211

Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black. 2016. Keep it SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image. In *Proceedings of the European Conference on Computer Vision (ECCV '16)*. Springer, Berlin, Germany, 561âĂŞ578.

Ernesto Brau and Hao Jiang. 2016. 3D Human Pose Estimation via Deep Learning from 2D Annotations. In *2016 Fourth International Conference on 3D Vision (3DV '16)*. 582–591. https://doi.org/10.1109/3DV.2016.84

Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2018. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. In *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '18)*. IEEE Computer Society, Washington, DC, USA.

João Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. 2016. Human Pose Estimation with Iterative Error Feedback. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '16)*. IEEE Computer Society, Washington, DC, USA, 4733–4742.

Ching-Hang Chen and Deva Ramanan. 2017. 3d human pose estimation= 2d pose estimation+ matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '17)*. IEEE Computer Society, Washington, DC, USA, 7035–7043.

Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. 2018a. Cascaded pyramid network for multi-person pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '18)*. IEEE Computer Society, Washington, DC, USA, 7103–7112.

Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. 2018b. Cascaded pyramid network for multi-person pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7103–7112.

CMU. 2019. CMU Graphics Lab Motion Capture Database. http://mocap.cs.cmu.edu/

Rishabh Dabral, Anurag Mundhada, Uday Kusupati, Safeer Afaque, Abhishek Sharma, and Arjun Jain. 2018. Learning 3D Human Pose from Structure and Motion. In *Proceedings of the European Conference on Computer Vision (ECCV'18)*, Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (Eds.). Springer International Publishing, 679–696.

Yuzhu Dong, Aishat Aloba, Sachin Paryani, Lisa Anthony, Neha Rana, and Eakta Jain. 2017. Adult2Child: Dynamic Scaling Laws to Create Child-like Motion. In *Proceedings of the Tenth International Conference on Motion in Games (MIG '17)*. ACM, New York, NY, USA, Article 13, 13:1–13:10 pages. https://doi.org/10.1145/3136457.3136460

Hao-Shu Fang*, Yuanlu Xu*, Wenguan Wang, Xiaobai Liu, and Song-Chun Zhu. 2018. Learning Pose Grammar to Encode Human Body Configuration for 3D Pose Estimation. In *AAAI Conference on Artificial Intelligence*.

Keith Grochow, Steven L. Martin, Aaron Hertzmann, and Zoran Popoviundefined. 2004. Style-Based Inverse Kinematics. In *ACM SIGGRAPH 2004 Papers (SIGGRAPH'04)*. Association for Computing Machinery, New York, NY, USA, 522âĂŞ531. https://doi.org/10.1145/1186562.1015755

Riza Alp Güler and Iasonas Kokkinos. 2019. HoloPose: Holistic 3D Human Reconstruction In-The-Wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'19)*. 10876–10886. https://doi.org/10.1109/CVPR.2019.01114

Riza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. 2018. DensePose: Dense Human Pose Estimation in the Wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '18)*. IEEE Computer Society, Washington, DC, USA, 7297–7306. https://doi.org/10.1109/CVPR.2018.00762

Semih Günel, Helge Rhodin, and Pascal Fua. 2018. What Face and Body Shapes Can Tell Us About Height. arXiv:cs.CV/1805.10355

Ikhsanul Habibie, Weipeng Xu, Dushyant Mehta, Gerard Pons-Moll, and Christian Theobalt. 2019. In the Wild Human Pose Estimation Using Explicit 2D Features and Intermediate 3D Representations. *arXiv preprint arXiv:1904.03289* (2019).

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*. 2961–2969.

Mir Rayat Imtiaz Hossain and James J Little. 2018. Exploiting temporal information for 3d human pose estimation. In *European Conference on Computer Vision*. Springer, 69–86.

Yinghao Huang, Federica Bogo, Christoph Lassner, Angjoo Kanazawa, Peter V Gehler, Javier Romero, Ijaz Akhter, and Michael J Black. 2017. Towards accurate marker-less human shape and pose estimation over time. In *2017 International Conference on 3D Vision (3DV)*. IEEE, 421–430.

Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. 2014. Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *IEEE Trans. Pattern Anal. Mach. Intell.* 36, 7 (July 2014), 1325–1339. https://doi.org/10.1109/TPAMI.2013.248

Sam Johnson and Mark Everingham. 2010. Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation. In *Proceedings of the British Machine Vision Conference (BMVC'10)*. doi:10.5244/C.24.12.

Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. 2018. End-to-End Recovery of Human Shape and Pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '18)*. IEEE Computer Society, Washington, DC, USA, 7122–7131. https://doi.org/10.1109/CVPR.2018.00744

Angjoo Kanazawa, Jason Y. JZhang, Panna Felsen, and Jitendra Malik. 2019. Learning 3D Human Dynamics from Video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'19)*.

Isinsu Katircioglu, Bugra Tekin, Mathieu Salzmann, Vincent Lepetit, and Pascal Fua. 2018. Learning latent representations of 3d human pose with deep neural networks. *International Journal of Computer Vision* 126, 12 (2018), 1326–1341.

Nikos Kolotouros, Georgios Pavlakos, Michael J. Black, and Kostas Daniilidis. 2019b. Learning to Reconstruct 3D Human Pose and Shape via Model-fitting in the Loop. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'19)*.

Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. 2019a. Convolutional Mesh Regression for Single-Image Human Shape Reconstruction. In *Proceeding of the IEEE/CVF Conference in Computer Vision and Pattern Recognition (CVPR'19)*.

Christoph Lassner, Javier Romero, Martin Kiefel, Federica Bogo, Michael J. Black, and Peter V. Gehler. 2017. Unite the People: Closing the Loop Between 3D and 2D Human Representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '17)*. IEEE Computer Society, Washington, DC, USA, 4704–4713. https://doi.org/10.1109/CVPR.2017.500

Kyoungoh Lee, Inwoong Lee, and Sanghoon Lee. 2018a. Propagating LSTM: 3D Pose Estimation Based on Joint Interdependency. In *Proceedings of the European Conference*

on Computer Vision (ECCV '18). Springer International Publishing, 123–141.

Kyungho Lee, Seyoung Lee, and Jehee Lee. 2018b. Interactive Character Animation by Learning Multi-objective Control. *ACM Trans. Graph.* 37, 6, Article 180 (Dec. 2018), 10 pages. https://doi.org/10.1145/3272127.3275071

Chen Li and Gim Hee Lee. 2019. Generating Multiple Hypotheses for 3D Human Pose Estimation with Mixture Density Network. *arXiv preprint arXiv:1904.05547* (2019).

Shuang Liang, Xiao Sun, and Yichen Wei. 2018. Compositional Human Pose Regression. *Computer Vision and Image Understanding* 176-177 (2018), 1–8.

Mude Lin, Liang Lin, Xiaodan Liang, Keze Wang, and Hui Cheng. 2017. Recurrent 3d pose sequence machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '17)*. IEEE Computer Society, Washington, DC, USA, 810–819.

Yebin Liu, Juergen Gall, Carsten Stoll, Qionghai Dai, Hans-Peter Seidel, and Christian Theobalt. 2013. Markerless Motion Capture of Multiple Characters Using Multiview Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 11 (Nov. 2013), 2720–2735. https://doi.org/10.1109/TPAMI.2013.47

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: A Skinned Multi-person Linear Model. *ACM Trans. Graph.* 34, 6 (Oct. 2015), 248:1–248:16. https://doi.org/10.1145/2816795.2818013

Diogo C. Luvizon, David Picard, and Hedi Tabia. 2018. 2D/3D Pose Estimation and Action Recognition Using Multitask Deep Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '18)*. IEEE Computer Society, Washington, DC, USA, 5137–5146.

Julieta Martinez, Rayat Hossain, Javier Romero, and James J. Little. 2017. A Simple Yet Effective Baseline for 3d Human Pose Estimation. In *Proceeding of the IEEE International Conference on Computer Vision (ICCV '17)*. 2659–2668. https://doi.org/10.1109/ICCV.2017.288

Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. 2017a. Monocular 3D Human Pose Estimation In The Wild Using Improved CNN Supervision. In *2017 Fifth International Conference on 3D Vision (3DV' 17)*. IEEE Computer Society, 506–516.

Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Mohamed Elgharib, Pascal Fua, Hans-Peter Seidel, Helge Rhodin, Gerard Pons-Moll, and Christian Theobalt. 2019. XNect: Real-time Multi-person 3D Human Pose Estimation with a Single RGB Camera. arXiv:cs.CV/1907.00837

Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. 2017b. VNect: Real-time 3D Human Pose Estimation with a Single RGB Camera. *ACM Trans. Graph.* 36, 4, Article 44 (July 2017), 44:1–44:14 pages. https://doi.org/10.1145/3072959.3073596

Francesc Moreno-Noguer. 2017. 3D Human Pose Estimation from a Single Image via Distance Matrix Regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '17)*. IEEE Computer Society, Washington, DC, USA, 1561–1570. https://doi.org/10.1109/CVPR.2017.170

Alejandro Newell, Kaiyu Yang, and Jia Deng. 2016. Stacked Hourglass Networks for Human Pose Estimation. In *Proceedings of the European Conference on Computer Vision (ECCV '18)*.

George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, and Kevin Murphy. 2017. Towards Accurate Multi-person Pose Estimation in the Wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '17)*. IEEE Computer Society, Washington, DC, USA.

Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. 2018a. Ordinal depth supervision for 3d human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7307–7316.

Georgios Pavlakos, Xiaowei Zhou, Konstantinos G. Derpanis, and Kostas Daniilidis. 2017. Coarse-to-Fine Volumetric Prediction for Single-Image 3D Human Pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '17)*. IEEE Computer Society, Washington, DC, USA, 1263–1272.

Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. 2018b. Learning to Estimate 3D Human Pose and Shape from a Single Color Image. In *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '18)*. IEEE Computer Society, Washington, DC, USA.

Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 2019. 3D human pose estimation in video with temporal convolutions and semi-supervised training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '19)*. IEEE Computer Society, Washington, DC, USA.

Dario Pavllo, David Grangier, and Michael Auli. 2018. QuaterNet: A Quaternion-based Recurrent Model for Human Motion. arXiv:cs.CV/1805.06485

Xue Bin Peng, Angjoo Kanazawa, Jitendra Malik, Pieter Abbeel, and Sergey Levine. 2018. Sfv: Reinforcement learning of physical skills from videos. In *SIGGRAPH Asia 2018 Technical Papers*. ACM, 178.

Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. 2012. Reconstructing 3D Human Pose from 2D Image Landmarks. In *Proceedings of the European Conference on Computer Vision (ECCV '12)*, Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 573–586.

Helge Rhodin, Mathieu Salzmann, and Pascal Fua. 2018a. Unsupervised Geometry-Aware Representation for 3D Human Pose Estimation. In *Proceedings of the European Conference on Computer Vision (ECCV '18)*.

Helge Rhodin, Jörg Spörri, Isinsu Katircioglu, Victor Constantin, Frédéric Meyer, Erich Müller, Mathieu Salzmann, and Pascal Fua. 2018b. Learning Monocular 3D Human Pose Estimation from Multi-view Images. In *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '18)*. IEEE Computer Society, Washington, DC, USA.

Nikolaos Sarafianos, Bogdan Boteanu, Bogdan Ionescu, and Ioannis A. Kakadiaris. 2016. 3D Human Pose Estimation: A Review of the Literature and Analysis of Covariates. *Comput. Vis. Image Underst.* 152, C (Nov. 2016), 1–20. https://doi.org/10.1016/j.cviu.2016.09.002

Toby Sharp. 2012. The Vitruvian Manifold: Inferring Dense Correspondences for One-shot Human Pose Estimation. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (CVPR '12)*. IEEE Computer Society, Washington, DC, USA, 103–110.

Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. 2013. Real-time Human Pose Recognition in Parts from Single Depth Images. *Commun. ACM* 56, 1 (Jan. 2013), 116–124. https://doi.org/10.1145/2398356.2398381

Leonid Sigal, Alexandru O. Balan, and Michael J. Black. 2009. HumanEva: Synchronized Video and Motion Capture Dataset and Baseline Algorithm for Evaluation of Articulated Human Motion. *International Journal of Computer Vision* 87, 1 (05 Aug 2009), 4. https://doi.org/10.1007/s11263-009-0273-6

Bugra Tekin, Pablo Márquez-Neila, Mathieu Salzmann, and Pascal Fua. 2017. Learning to fuse 2d and 3d image cues for monocular body pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV '17)*. IEEE Computer Society, Washington, DC, USA, 3941–3950.

Bugra Tekin, Artem Rozantsev, Vincent Lepetit, and Pascal Fua. 2016. Direct prediction of 3d body poses from motion compensated sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '16)*. IEEE Computer Society, Washington, DC, USA, 991–1000.

Denis Tome, Chris Russell, and Lourdes Agapito. 2017. Lifting from the Deep: Convolutional 3D Pose Estimation from a Single Image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '17)*. IEEE Computer Society, Washington, DC, USA, 2500–2509.

Alexander Toshev and Christian Szegedy. 2014. DeepPose: Human Pose Estimation via Deep Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*. IEEE Computer Society, Washington, DC, USA, 1653–1660. https://doi.org/10.1109/CVPR.2014.214

Ruben Villegas, Jimei Yang, Duygu Ceylan, and Honglak Lee. 2018. Neural kinematic networks for unsupervised motion retargetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '18)*. IEEE Computer Society, Washington, DC, USA, 8639–8648.

Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popović. 2008. Articulated Mesh Animation from Multi-view Silhouettes. *ACM Trans. Graph.* 27, 3, Article 97 (Aug. 2008), 97:1–97:9 pages. https://doi.org/10.1145/1360612.1360696

Keze Wang, Liang Lin, Chenhan Jiang, Chen Qian, and Pengxu Wei. 2019. 3D Human Pose Machines with Self-supervised Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* (2019).

Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. 2016. Convolutional Pose Machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '16)*. IEEE Computer Society, Washington, DC, USA, 4724–4732. https://doi.org/10.1109/CVPR.2016.511

Xiaolin Wei, Peizhao Zhang, and Jinxiang Chai. 2012. Accurate Realtime Full-body Motion Capture Using a Single Depth Camera. *ACM Trans. Graph.* 31, 6, Article 188 (Nov. 2012), 188:1–188:12 pages. https://doi.org/10.1145/2366145.2366207

Weipeng Xu, Avishek Chatterjee, Michael Zollhöfer, Helge Rhodin, Dushyant Mehta, Hans-Peter Seidel, and Christian Theobalt. 2018. MonoPerfCap: Human Performance Capture From Monocular Video. *ACM Trans. Graph.* 37, 2, Article 27 (May 2018), 27:1–27:15 pages. https://doi.org/10.1145/3181973

Yuanlu Xu, Song-Chun Zhu, and Tony Tung. 2019. DenseRaC: Joint 3D Pose and Shape Estimation by Dense Render-and-Compare. arXiv:cs.CV/1910.00116

Wei Yang, Wanli Ouyang, Xiaolong Wang, Jimmy Ren, Hongsheng Li, and Xiaogang Wang. 2018. 3D Human Pose Estimation in the Wild by Adversarial Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '18)*. IEEE Computer Society, Washington, DC, USA, 5255–5264. https://doi.org/10.1109/CVPR.2018.00551

Mao Ye and Ruigang Yang. 2014. Real-Time Simultaneous Pose and Shape Estimation for Articulated Objects Using a Single Depth Camera. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*. IEEE Computer Society, Washington, DC, USA, 2353–2360. https://doi.org/10.1109/CVPR.2014.301

Anastasios Yiannakides, Andreas Aristidou, and Yiorgos Chrysanthou. 2019. Real-time 3D Human Pose and Motion Reconstruction from Monocular RGB Videos. *Comput. Animat. Virtual Worlds* 30, 3-4 (May 2019).

Yusuke Yoshiyasu, Ryusuke Sagawa, Ko Ayusawa, and Akihiko Murai. 2018. Skeleton Transformer Networks: 3D Human Pose and Skinned Mesh from Single RGB Image. arXiv:cs.CV/1812.11328

Jason Y. Zhang, Panna Felsen, Angjoo Kanazawa, and Jitendra Malik. 2019. Predicting 3D Human Dynamics from Video. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'19)*.

Zerong Zheng, Tao Yu, Yixuan Wei, Qionghai Dai, and Yebin Liu. 2019. DeepHuman: 3D Human Reconstruction from a Single Image. In *arXiv*.

Xingyi Zhou, Qi-Xing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. 2017. Towards 3d human pose estimation in the wild: a weakly-supervised approach. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV '17)*. IEEE Computer Society, Washington, DC, USA, 398–407.

Xingyi Zhou, Xiao Sun, Wei Zhang, Shuang Liang, and Yichen Wei. 2016. Deep kinematic pose regression. In *Proceedings of the European Conference on Computer Vision (ECCV '16)*. Springer International Publishing, 186–201.

Xiaowei Zhou, Menglong Zhu, Georgios Pavlakos, Spyridon Leonardos, Konstantinos G. Derpanis, and Kostas Daniilidis. 2018. MonoCap: Monocular Human Motion Capture using a CNN Coupled with a Geometric Prior. *IEEE Trans. Pattern Anal. Mach. Intell.* (2018). https://doi.org/10.1109/tpami.2018.2816031

Yi Zhou, Connelly Barnes, Lu Jingwan, Yang Jimei, and Li Hao. 2019. On the Continuity of Rotation Representations in Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'19)*.

A NETWORK ARCHITECTURE

In this part we describe the layers of the components in our architecture (E_S, E_Q and D) and list them in Table 4.

In both E_S and E_Q there are three main parts: channel-expansion, information-fusion and channel-shrink. In the expansion and shrink parts, we use convolutional blocks, where the convolutions have kernel size 1 and stride 1. In the information-fusion part, for E_Q, we apply three parallel convolutional blocks, where each has a different kernel size, and then use adaptive average pooling to ensure that the temporal length of the output sequence is equal to the input one. For E_s, we use sequential convolution blocks and then collapse all the temporal information into a single vector with adaptive max pooling.

Our discriminator D is a linear component (similarly to Kanazawa et al. [2018]), with an output value between 0 and 1, containing two convolution layers and one fully connected layer.

In each convolution block, we use BatchNorm, LReLU, Dropout. The kernel width and the stride are denoted by k and s, respectively, and the number of input and output channels is given in the rightmost column.

Name		Layers	k	s	in/out
E_Q		Conv + BatchNorm + LReLU + Dropout	1	1	34/2048
	→	Conv + BatchNorm + LReLU + Dropout + Adap AP	5	3	2048/2048
	→	Conv + BatchNorm + LReLU + Dropout + Adap AP	3	1	2048/2048
	→	Conv + BatchNorm + LReLU + Dropout + Adap AP	1	1	2048/2048
		Conv	1	1	1024/69
E_S		Conv + BatchNorm + LReLU + Dropout	1	1	34/2048
		Conv + BatchNorm + LReLU + Dropout	5	1	2048/2048
	↓	Conv + BatchNorm + LReLU + Dropout	3	1	2048/2048
		Conv + BatchNorm + LReLU + Dropout	1	1	2048/2048
		Adaptive MP	-	-	-
		Conv	1	1	2048/9
D		Conv	1	1	4/16
	↓	Conv	1	1	16/16
		Linear	–	-	16/1

Table 4. MotioNet structure. → denotes parallel convolutions while ↓ denotes sequential ones.