Structured Prediction Helps 3D Human Motion Modelling

Emre Aksan[*] Manuel Kaufmann[*] Otmar Hilliges
Department of Computer Science, ETH Zürich
{firstname.lastname}@inf.ethz.ch

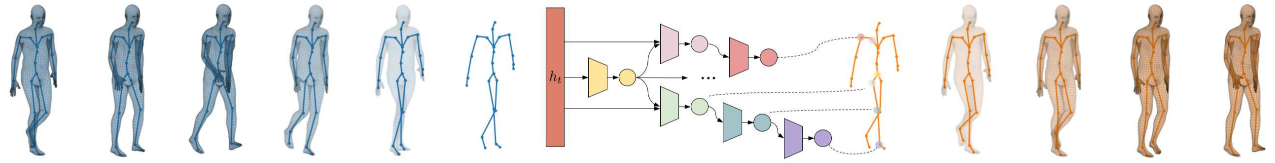

Figure 1: We introduce a structured prediction layer (SPL) to the task of 3D human motion modelling. The SP-layer explicitly decomposes the pose into individual joints and can be interfaced with a variety of baseline architectures. We show that on H3.6M and a recent, much larger dataset, AMASS, a variety of baseline models benefit when augmented with an SP-layer.

Abstract

Human motion prediction is a challenging and important task in many computer vision application domains. Existing work only implicitly models the spatial structure of the human skeleton. In this paper, we propose a novel approach that decomposes the prediction into individual joints by means of a structured prediction layer that explicitly models the joint dependencies. This is implemented via a hierarchy of small-sized neural networks connected analogously to the kinematic chains in the human body as well as a joint-wise decomposition in the loss function. The proposed layer is agnostic to the underlying network and can be used with existing architectures for motion modelling. Prior work typically leverages the H3.6M dataset. We show that some state-of-the-art techniques do not perform well when trained and tested on AMASS, a recently released dataset 14 times the size of H3.6M. Our experiments indicate that the proposed layer increases the performance of motion forecasting irrespective of the base network, joint-angle representation, and prediction horizon. We furthermore show that the layer also improves motion predictions qualitatively. We make code and models publicly available at `https://ait.ethz.ch/projects/2019/spl`.

1. Introduction

Modelling of human motion over time has a number of applications in activity recognition, human computer interaction, human detection and tracking, and image-based pose estimation in the context of robotics or self-driving vehicles. Humans have the ability to forecast the sequence of poses over short-term horizons with high accuracy and can imagine probable motion over arbitrary time scales. Despite recent progress in data-driven modelling of human motion [7, 8, 14, 20, 25, 33], this task remains difficult for machines.

The difficulty of the task is manifold. First, human motion is highly dynamic, non-linear and over time becomes a stochastic sequential process with a high degree of inherent uncertainty. Humans leverage strong structural and temporal priors about continuity and regularity in natural motion. However, these are hard to model algorithmically due to i) the inter-dependencies between joints and ii) the influence of high-level activities on the motion sequences (e.g., transition from walking to jumping). In fact many recent approaches forgo explicit modelling of human motion [14] in favor of pure data-driven models [8, 20, 25].

Initial Deep Learning-based motion modelling approaches have focused on recurrent neural networks (RNNs) [8, 7, 14], using curriculum learning schemes to increase robustness to temporal drift. Martinez *et al*. [20] have shown that a simple running-average provides a surprisingly difficult to beat baseline in terms of Euler angle error. Following this, sequence-to-sequence models trained in an auto-regressive fashion have been proposed [20], sometimes using adversarial training to address the drift problem in long-term predictions [33]. Pavllo *et al*. [25] study the impact of joint angle representation and show that a quaternion-based parameterization improves short-term predictions.

However, it has been observed that quantitative performance does not always translate to qualitatively meaningful predictions [20, 25]. Furthermore, the H3.6M benchmark is becoming saturated, limiting progress. This leads to the two main research questions studied in this work: i) How to measure accuracy of pose predictions in a meaningful way such that low errors corresponds to good qualitative results and how to improve this performance? ii) How to exploit spatial structure of the human skeleton for better predictions?

With respect to i) we note that much of the literature relies on the H3.6M [12] dataset and an Euler angle based metric as performance measure, evaluated on a limited number of

[*] The first two authors contributed equally.

test sequences. While enabling initial exploration of the task, the dataset is limited in size (roughly 3 hours from 210 sequences) and in diversity of activities and poses, which contributes to a saturation effect in terms of performance. In this paper we show that existing techniques do not scale well when trained on larger and more diverse datasets. To this end, we leverage the recently released AMASS dataset [19], itself consisting of multiple smaller motion datasets, offering many more samples (14x over H3.6M) and a wider range of activities. To further unpack the performance of motion modelling techniques, we introduce several evaluation metrics to the task of human motion prediction.

Our main technical contribution is a novel structured prediction layer (SPL) that addresses our second research question. We leverage the compositional structure of the human skeleton by explicitly decomposing the pose into individual joints. The SP-layer models the structure of the human skeleton and hence the spatial dependencies between joints. This is achieved via a hierarchy of small-sized neural networks that are connected analogously to the kinematic chains of the human skeleton. Each node in the graph receives information about the parent node's prediction and thus information is propagated along the kinematic chain. We furthermore introduce a joint-wise decomposition of the loss function as part of SPL. The proposed layer is agnostic to the underlying network and can be used in combination with most previously proposed architectures.

We show experimentally that introducing this layer to existing approaches improves performance of the respective method. The impact is most pronounced on the larger and more challenging AMASS dataset. This indicates that our approach is indeed a step towards successfully exploiting spatial priors in human motion modelling and in turn allows recurrent models to capture temporal coherency more effectively. We thoroughly evaluate the SP-layer on H3.6M and AMASS. On AMASS, for any base model, any metric, and any input representation, it is beneficial to use the SP-layer. Furthermore, even simple architectures that are outperformed by a zero-velocity baseline [20] perform competitive if paired with the SP-layer.

In summary, we contribute: i) An in-depth analysis of state-of-the-art motion modelling methods and their evaluation. ii) A new benchmark and evaluation protocol on the recent, much larger AMASS dataset. iii) A novel prediction layer, incorporating structural priors. iv) A thorough evaluation of the SP-layer's impact on motion modelling in combination with several base models.

2. Related Work

We briefly review the most related literature on human motion modelling focusing on Deep Learning for brevity.

Deep recurrent models Early work makes use of specialized Deep Belief Networks for motion modelling [30], whereas more recent works leverage recurrent architectures. For example, Fragkiadaki *et al.* [7] propose the Encoder-Recurrent-Decoder (ERD) framework, which maps pose data into a latent space where it is propagated through time via an LSTM cell. The prediction at time step t is fed back as the input for time step $t + 1$. This scheme quickly leads to error accumulation and hence catastrophic drift over time. To increase robustness, Gaussian noise is added during training. While alleviating the drift problem, this training scheme is hard to fine-tune. Quantitative and qualitative evaluations are performed on the publicly available H3.6M dataset [12], with a joint angle data representation using the exponential map (also called angle-axis). The joint-wise Euclidean distance on the Euler angles is used as the evaluation metric. Most of the follow-up work adheres to this setting.

Inspired by [7], Du *et al.* [6] have recently proposed to combine a three-layer LSTM with bio-mechanical constraints encoded into the loss function for pedestrian pose and gait prediction. Like [6], we also incorporate prior knowledge into our network design, but do so through a particular design of the output layer rather than enforcing physical constraints in the loss function. Similar in spirit to [7], Ghosh *et al.* [8] stabilize forecasting for long-term prediction horizons via application of dropouts on the input layer of a denoising autoencoder. In this work we focus on short-term predictions, but also apply dropouts directly on the inputs to account for noisy predictions of the model at test time. Contrary to [8], our model can be trained end-to-end.

Martinez *et al.* [20] employ a sequence-to-sequence architecture using a single layer of GRU cells [4]. The model is trained auto-regressively, using its own predictions during training. A residual connection on the decoder leads to smoother and improved short-term predictions. Martinez *et al.* also show that simple running-average baselines are surprisingly difficult to beat in terms of the Euler angle metric. The currently best performance on H3.6M is reported by Wang *et al.* [33]. They also use a sequence-to-sequence approach trained with an adversarial loss to address the drift-problem and to create smooth predictions. Highlighting some of the issues with the previously used L_2 loss, [33] propose a more meaningful geodesic loss.

In this work we show that sequence-to-sequence models, despite good performance on H3.6M, do not fare as well on the larger, more diverse AMASS dataset. Although augmenting them with our SP-layer boosts their performance, they are outperformed by a simple RNN that uses the same SP-layer. To better characterize motion modelling performance we furthermore introduce several new evaluation metrics.

Structured Prediction Jain *et al.* [14] propose to explicitly model structural information by automatically converting an st-graph into an RNN (S-RNN). The skeleton is divided into 5 major clusters, whose interactions are then manually encoded into an st-graph. Our model is also structure-aware. However, our approach does not require a coarse subdivision of joints and does not require manual definition of st-graphs. Moreover, our layer is agnostic to the underlying network and can be interfaced with most existing architectures.

Bütepage *et al.* [2] propose to encode poses with a hierarchy of dense layers following the kinematic chain starting from the end-effectors (dubbed H-TE), which is similar to our SP-layer. In contrast to this work, H-TE operates on the input rather than the output, and has only been demonstrated with non-recurrent networks when using 3D positions to parameterize the poses.

Structure-aware network architectures have also been used in 3D pose estimation from images [16, 29, 21, 17, 31]. [17] and [31] both learn a structured latent space. [21] exploit structure only implicitly by encoding the poses into distance matrices which then serve as inputs and outputs of the network. [16] and [29] are closest to our work as they explicitly modify the network to account for skeletal structure, either via the loss function [29], or using a sequence of LSTM cells for each joint in the skeleton [16]. [16] introduces many new layers into the architecture and needs hyper-parameter tuning to be most effective. In contrast, our proposed SP-layer is simple to implement and train. We show that it improves performance of several baseline architectures out-of-the-box.

Parameterizations Most work parameterizes joint angles as exponential maps relative to each joint's parent. Pavllo *et al.* [25] show results competitive with the state of the art using quaternions. Their model, QuaterNet, consists of 2 layers of GRU cells and similar to [20] uses a skip connection. The use of quaternions allows for integration of a differentiable forward kinematics layer, facilitating loss computation in the form of Euclidean distance of 3D joint positions. For short-term predictions, QuaterNet directly optimizes for the Euler-angle based metric as introduced by [7]. We show that QuaterNet also benefits from augmentation with our SP-layer, indicating that SPL is independent of the underlying joint angle representation.

Bütepage *et al.* [2, 3] and Holden *et al.* [10] convert the data directly to 3D joint positions. These works do not use recurrent structures, which necessitates the extraction of fixed-size, temporal windows for training. [2] and [10] focus on learning of latent representations, which are shown to be helpful for various tasks, such as denoising, forecasting, or motion generation along a given trajectory [9]. [3] extends [2] by applying a conditional variational autoencoder (VAE) to the task of online motion prediction in human-robot interactions. We use the positional representation of human poses to compute an informative metric of the prediction quality. However, for learning we use joint angles since they encode symmetries better and are inherently bone-length invariant.

3. Method

The goal of our work is to provide a general solution to the problem of human motion modelling. To this end we are motivated by the observation that human motion is strongly regulated by the spatial structure of the skeleton. However, integrating this structure into deep neural network architectures has so far not yielded better performance than architectures that only model temporal dependencies explicitly. In this section we outline a novel structured prediction layer (SPL) that explicitly captures the spatial connectivity. The layer is designed to be agnostic to the underlying network. We empirically show in Sec. 5 and 6 that it improves the performance of a variety of existing models irrespective of the dataset or the data representation used.

3.1. Problem Formulation

A motion sample can be considered as a sequence $\boldsymbol{X} = \{\boldsymbol{x}_1 \dots \boldsymbol{x}_T\}$ where a frame $\boldsymbol{x}_t \in \mathbb{R}^N$ at time-step t denotes the N-dimensional body pose. N depends on the number of joints in the skeleton, K, and the size M of the per-joint representation (angle-axis, rotation matrices, quaternions, or 3D positions), i.e. $N = K \cdot M$.

Due to their temporal nature, motion sequences are often modelled with auto-regressive approaches. Such models factorize the joint probability of a motion sequence as a product of conditionals as follows:

$$p_\theta(\boldsymbol{X}) = \prod_{t=1}^{T} p_\theta(\boldsymbol{x}_t \mid \boldsymbol{x}_{1:t-1}) \tag{1}$$

where the joint distribution is parameterized by θ. At each time step t, the next pose is predicted given the past poses.

While this auto-regressive setting explicitly models the temporal dependencies, the spatial structure is treated only implicitly. In other words, given a pose vector $\boldsymbol{x}_t$, the model must predict the whole pose vector $\boldsymbol{x}_{t+1}$ at the next time step. This assumes that joints are independent from each other given a particular context (i.e., a neural representation of the past frames). However, the human body is composed of hierarchical joints and the kinematic chain introduces spatial dependencies between them.

3.2. Structured Prediction Layer

To address this shortcoming, we propose a novel structured prediction layer (SPL). This is formed by decomposing the model prediction into individual joints. This decomposition is guided by the spatial prior of the human kinematic chain, depicted in Fig. 2. Formally, $\boldsymbol{x}_t \in \mathbb{R}^N$ is a concatenation of K joints $\boldsymbol{x}_t^{(k)} \in \mathbb{R}^M$:

$$\boldsymbol{x}_t = [\boldsymbol{x}_t^{(hip)}, \boldsymbol{x}_t^{(spine)} \dots \boldsymbol{x}_t^{(lwrist)}, \boldsymbol{x}_t^{(lhand)}]$$

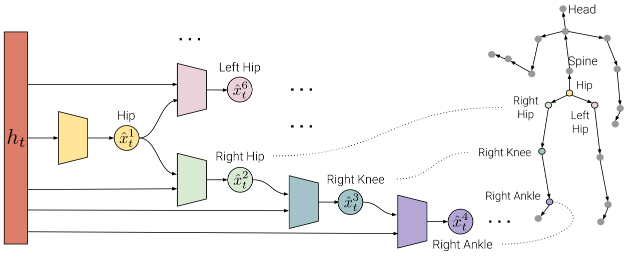

Figure 2: **SPL overview**. Given the context $\boldsymbol{h}_t$ of past frames, joint predictions $\hat{\boldsymbol{x}}_t^{(k)}$ are made hierarchically by following the kinematic chain defined by the underlying skeleton. Only a subset of joints is visualized for clarity.

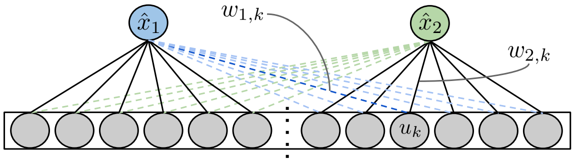

Figure 3: **Difference between dense and SP-layer** with 2 joints. When all dashed weights are zero, a dense hidden layer is equivalent to a SP-layer that ignores the hierarchy. In a dense layer, the hidden unit u_k is connected to all joints via $w_{1,k}$ and $w_{2,k}$. Hence, the gradient $\partial L/\partial u_k$ is affected by both joints, whereas in SPL only $w_{2,k}$ contributes by design.

To interface with existing architectures, the SP-layer takes a context representation $\boldsymbol{h}_t$ as input. Here, $\boldsymbol{h}_t$ is assumed to summarize the motion sequence until time t. Without loss of generality, we assume this to be a hidden RNN state or its projection. While existing work typically leverages several dense layers to predict the N-dimensional pose vector $\boldsymbol{x}_t$ from $\boldsymbol{h}_t$, our SP-layer predicts each joint individually with separate smaller networks:

$$p_\theta(\boldsymbol{x}_t) = \prod_{k=1}^{K} p_\theta(\boldsymbol{x}_t^{(k)} \mid \text{parent}(\boldsymbol{x}_t^{(k)}), \boldsymbol{h}_t) \qquad (2)$$

where parent$(\boldsymbol{x}_t^{(k)})$ extracts the parent of the k-th joint. Importantly, the full body pose $\boldsymbol{x}_t$ is predicted by following the skeletal hierarchy in Fig. 2 as follows:

$$p_\theta(\boldsymbol{x}_t) = p_\theta(\boldsymbol{x}_t^{(hip)} \mid \boldsymbol{h}_t) p_\theta(\boldsymbol{x}_t^{(spine)} \mid \boldsymbol{x}_t^{(hip)}, \boldsymbol{h}_t) \cdots \quad (3)$$

In this formulation each joint receives information about its own configuration and that of the immediate parent both explicitly, through the conditioning on the parent joint's prediction, and implicitly via the context $\boldsymbol{h}_t$. The joint probability of Eq. 1 is further factorized in the spatial domain:

$$p_\theta(\boldsymbol{X}) = \prod_{t=1}^{T} \prod_{k=1}^{K} p_\theta(\boldsymbol{x}_t^{(k)} \mid \text{parent}(\boldsymbol{x}_t^{(k)}), \boldsymbol{h}_t) \qquad (4)$$

The benefit of this structured prediction approach is twofold. First, the proposed factorization allows for integration of a structural prior in the form of a hierarchical architecture where each joint is modelled by a different network. This allows the model to learn dedicated representations per joint and hence saves model capacity. Second, analogous to message passing, each parent propagates its prediction to the child joints, allowing for more precise local predictions because the joint has access to the information it depends on (i.e., the parent's prediction).

In our experiments (cf. Sec. 5 and 6) we show that this layer improves the prediction performance of a diverse set of underlying architectures across many settings and metrics. One potential reason for why this is the case can be found in the resulting network structure and its implications on network training. Fig. 3 compares our structured approach with the traditional one-shot prediction using a dense layer. Because the per-joint decomposition leads to many small separate networks, we can think of an SP-layer as a dense layer where some connections have been set to zero explicitly by leveraging domain knowledge. This decomposition changes the gradients w.r.t. the units in the hidden layer, which are now only affected by the gradients coming from the joint hierarchy that they model. In the traditional setting, the error computed as an average over all joints can easily be distributed over all network weights in an arbitrary fashion.

3.3. Per-joint Loss

We additionally propose to perform a similar decomposition in the objective function that leads to further improvements. The training objective is often a metric in Euclidean space between ground-truth poses $\boldsymbol{x}_t$ and predictions $\hat{\boldsymbol{x}}_t$:

$$\mathcal{L}(\boldsymbol{X}, \hat{\boldsymbol{X}}) = \frac{1}{T \cdot N} \sum_{t=1}^{T} f(\boldsymbol{x}_t, \hat{\boldsymbol{x}}_t) \qquad (5)$$

where f is a loss function such as an ℓ_p norm. The loss f is calculated on the entire pose vector and averaged across the temporal and spatial domain. In our work, we use a slightly modified version that preserves joint integrity:

$$\mathcal{L}(\boldsymbol{X}, \hat{\boldsymbol{X}}) = \sum_{t=1}^{T} \sum_{k=1}^{K} f(\boldsymbol{x}_t^{(k)}, \hat{\boldsymbol{x}}_t^{(k)}) \qquad (6)$$

where the loss f is first calculated on every joint and then summed up to calculate the loss for the entire motion sequence. In this work we use the MSE for f, but the formulation allows for an easy adaptation of domain-specific losses such as the *geodesic distance* proposed by [33].

4. Human Motion Modelling

We now evaluate our SP-layer on the task of human motion modelling. We perform our experiments on two datasets and three different underlying architectures which use three different data representations. In the following we explain the datasets and models in more detail.

4.1. Datasets

For ease of comparison to the state of the art we first report results from the H3.6M dataset. We follow the same experiment protocol used in [14, 20].

Given the small size of H3.6M and the reported variance of test results [25], we propose to use the recently introduced AMASS dataset [19] for the motion modelling task. We downloaded the dataset from [11] as the data from [19] has not yet been released at the time of this writing. AMASS is composed of publicly available databases, *e.g.* the CMU Mocap database [5] or HumanEva [26] and uses the SMPL model [18] to represent motion sequences. The dataset contains $8'593$ sequences, which comprise a total of $9'084'918$ frames sampled at 60 Hz. This is roughly equivalent to 42 hours of recording, making AMASS about 14 times bigger than H3.6M ($632'894$ frames at 50 Hz).

We split the AMASS dataset into training, validation and test splits consisting of roughly 90%, 5% and 5% of the samples, respectively. Similar to the H3.6M protocol, the input sequences are 2 seconds (120 frames) and the target sequences are 400-ms (24 frames) long. The H3.6M benchmarks use a total of 120 test samples across 15 categories. This is a relatively small test set and it has been reported to cause high variance [24]. In our H3.6M experiments we use this setting to ensure fair comparison. However, on AMASS we use every frame in the test split by shifting a 2-second window over the motion sequences, which extracts $3'304$ test samples. H3.6M and AMASS model the human skeleton with 21 and 15 major joints, respectively. We implement separate SP-layers corresponding to the underlying skeleton.

4.2. Models

The modular nature of our SP-layer allows for flexible deployment with a diverse set of base models. In our experiments, we test the layer with the following three representative architectures proposed in the literature. To ease experimentation with SPL and other base architectures, we make all code and pre-trained models available at `https://ait.ethz.ch/projects/2019/spl`.

Seq2seq is a model proposed by Martinez *et al.* [20], consisting of a single layer of GRU cells. It contains a residual connection between the inputs and predictions. Input poses are represented as exponential maps.

QuaterNet uses a quaternion representation instead [24, 25]. The model augments RNNs with quaternion based normalization and regularization operations. Similarly, the residual connection from inputs to outputs is implemented via the quaternion product. In our experiments, we replace the final linear output layer with our SP-layer and keep the remaining setup intact.

RNN uses a single layer recurrent network to calculate the context $\boldsymbol{h}_t$, which we feed to our SP-layer. In contrast to the Seq2seq and QuaterNet settings, we represent poses via rotation matrices. To account for the error accumulation problem at test time [7, 8, 14], we apply dropout directly on the inputs. This architecture is similar to the ERD [7] but is additionally augmented with the residual connection of [20].

In the SP-layer, each joint is modelled with only one small hidden layer (64 or 128 units) followed by a ReLU activation and a linear projection to the joint prediction $\hat{\boldsymbol{x}}_t^{(k)} \in \mathbb{R}^M$. We experiment with different hierarchical configurations in SPL (cf. Sec. 6.3) where following the true kinematic chain performed best. Some models benefit from inputting all parent joints in the kinematic chain compared to using only the immediate parent. Note that we changed existing Seq2seq and QuaterNet models only as much as required to integrate them with SPL. To ensure a fair comparison we fine-tune hyper-parameters like learning rate, batch size and hidden layer units. See appendix Sec. 8.1 for details.

5. Evaluation on H3.6M Dataset

In our first set of comparisons we baseline the proposed SP-layer on the H3.6M dataset using the Euler angle metric as is common practice in the literature.

5.1. Metrics

Euler angles Let $\boldsymbol{w} = \theta \boldsymbol{a}$ denote a rotation of angle θ around the unit axis $\boldsymbol{a} \in \mathbb{R}^3$. $\boldsymbol{w}$ is the angle-axis (or exponential map) representation of a single joint angle. The Euler angles are extracted from $\boldsymbol{w}$ by first converting it into a rotation matrix $\boldsymbol{R} = \exp(\boldsymbol{w})$ using Rodrigues' formula and then computing the angles $\boldsymbol{\alpha} = (\alpha_x, \alpha_y, \alpha_z)$ following [27]. This assumes that $\boldsymbol{R}$ follows the *z*-*y*-*x* order. Furthermore, as noted by [27], there exist always two solutions for $\boldsymbol{\alpha}$, from which [14] picks the one that leads to the least amount of rotation. The Euler angle metric for time step t is then

$$L_{eul}(t) = \frac{1}{|\mathcal{X}_{test}|} \sum_{\boldsymbol{x}_t \in \mathcal{X}_{test}} \sqrt{\sum_k (\boldsymbol{\alpha}_t^{(k)} - \hat{\boldsymbol{\alpha}}_t^{(k)})^2} \qquad (7)$$

where $\boldsymbol{\alpha}_t^{(k)}$ are the predicted Euler angles of joint k at time t. $\mathcal{X}_{test}$ is defined by [14] and comprises of 120 sequences.

5.2. Results

Tab. 1 summarizes the relative performances of models with and without the SP-layer on the H3.6M dataset and compares them to the state of the art. The publicly available

	Walking				Eating				Smoking				Discussion			
milliseconds	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
LSTM-3LR [7]	0.77	1.00	1.29	1.47	0.89	1.09	1.35	1.46	1.34	1.65	2.04	2.16	1.88	2.12	2.25	2.23
SRNN [14]	0.81	0.94	1.16	1.30	0.97	1.14	1.35	1.46	1.45	1.68	1.94	2.08	1.22	1.49	1.83	1.93
Zero-Velocity [20]	0.39	0.68	0.99	1.15	0.27	0.48	0.73	0.86	0.26	0.48	0.97	0.95	0.31	0.67	0.94	1.04
AGED [33]	0.22	0.36	0.55	0.67	**0.17**	**0.28**	**0.51**	**0.64**	0.27	**0.43**	**0.82**	**0.84**	0.27	**0.56**	**0.76**	**0.83**
Seq2seq-sampling-sup [20]	0.28	0.49	0.72	0.81	0.23	0.39	0.62	0.76	0.33	0.61	1.05	1.15	0.31	0.68	1.01	1.09
Seq2seq-sampling-sup-SPL	0.23	0.37	**0.53**	**0.61**	0.20	0.32	0.52	0.67	0.26	0.48	0.92	0.90	0.29	0.63	0.90	0.99
Seq2seq-sampling [20]	0.27	0.47	0.70	0.78	0.25	0.43	0.71	0.87	0.33	0.61	1.04	1.19	0.31	0.69	1.03	1.12
Seq2seq-sampling-SPL	0.23	0.38	0.58	0.67	0.20	0.32	0.52	0.66	0.26	0.48	0.92	0.90	0.30	0.64	0.91	0.99
QuaterNet [25]	**0.21**	**0.34**	0.56	0.62	0.20	0.35	0.58	0.70	**0.25**	0.47	0.93	0.90	**0.26**	0.60	0.85	0.93
QuaterNet-SPL	0.22	0.35	0.54	**0.61**	0.20	0.33	0.55	0.68	**0.25**	0.47	0.91	0.88	**0.26**	0.59	0.84	0.91
RNN	0.30	0.48	0.78	0.89	0.23	0.36	0.57	0.72	0.26	0.49	0.97	0.95	0.31	0.67	0.95	1.03
RNN-SPL	0.26	0.40	0.67	0.78	0.21	0.34	0.55	0.69	0.26	0.48	0.96	0.94	0.30	0.66	0.95	1.05

Table 1: **H3.6M results** for the commonly used *walking*, *eating*, *smoking*, and *discussion* activities across different prediction horizons. Values correspond to the Euler angle metric measured *at* the given time. "Seq2seq-sampling" and "Seq2seq-sampling-sup" models correspond to "Residual unsup. (MA)" and "Residual sup. (MA)" models in [20], respectively. Note the relative performance improvement for each base model when augmented with our SP-layer.

Seq2seq [20] and QuaterNet [25] models are augmented with our SP-layer, but we otherwise follow the original training and evaluation protocols of the respective baseline model.

Using the SP-layer improves the Seq2seq performance significantly and achieves state-of-the-art performance in the *walking* category. Similarly, SPL yields the best performance with QuaterNet in short-term *smoking* and *discussion* motions and marginally outperforms the vanilla QuaterNet in most categories or is comparative to it. While our SP-layer also boosts the performance of the RNN model in *walking*, *eating* and *smoking* motion categories, performance remains similar for *discussion*.

We follow the same evaluation setting as in previous work for direct comparability. It is noteworthy to mention that the evaluation metrics reported on H3.6M exhibit high variance due to the small number of test samples [24] and low errors do not always correspond to good qualitative results [20].

6. AMASS: A New Benchmark

In this section we evaluate the baseline methods and our SP-layer on the large-scale AMASS dataset, detailed in Sec. 4.1. The diversity and large amount of motion samples in AMASS increase both the task's complexity and the reliability of results due to a larger test set. In addition to proposing a new evaluation setting for motion modelling we suggest usage of a more versatile set of metrics for the task.

6.1. Metrics

So far, motion prediction has been benchmarked on H3.6M using the Euclidean distance between target and predicted Euler angles [14, 20, 25, 33]. Numbers are usually reported per action at certain time steps averaged over 8 samples [14]. Unfortunately, Euler angles have twelve different conventions (not counting the fact that each of these can be defined using intrinsic or extrinsic rotations), which makes the practical implementation of this metric error-prone.

For a more precise analysis we introduce additional metrics from related pose estimation areas [28, 32, 34]. In order to increase the robustness we furthermore suggest to i) sum *until* time step t rather than report the metric *at* time step t, ii) use more test samples covering a larger portion of the test data set and iii) evaluate the models with complementary metrics. Note that we do not train the models on these metrics; they only serve as evaluation criteria at test time.

Joint angle difference To circumvent the potential source of error in the Euler angle metric, we propose using another angle-based metric following [11, 32]. This metric computes the angle of the rotation required to align the predicted joint with the target joint. Unlike L_{eul}, this metric is independent of how rotations are parameterized. It is furthermore similar to the geodesic loss proposed by [33]. Let $\hat{\boldsymbol{R}}$ be the predicted joint angle for a given joint, parameterized as a rotation matrix, and the respective target rotation $\boldsymbol{R}$. The difference in rotation can be computed as $\tilde{\boldsymbol{R}} = \hat{\boldsymbol{R}}\boldsymbol{R}^T$, from which we construct the metric at time step t as follows:

$$L_{angle}(t) = \frac{1}{|\mathcal{X}_{test}|} \sum_{\boldsymbol{x}_t \in \mathcal{X}_{test}} \frac{1}{K} \sum_{k} \left\| \log\left(\tilde{\boldsymbol{R}}_t^{(k)}\right) \right\|_2 \quad (8)$$

where $\tilde{\boldsymbol{R}}_t^{(k)}$ is the rotation matrix of joint k at time t. In contrast to L_{eul} we compute the loss on global joint angles by first unrolling the kinematic chain before computing L_{angle}.

	Euler				Joint Angle				Positional				PCK (AUC)			
milliseconds	100	200	300	400	100	200	300	400	100	200	300	400	100	200	300	400
Zero-Velocity [20]	1.91	5.93	11.36	17.78	0.37	1.22	2.44	3.94	0.14	0.48	0.96	1.54	0.86	0.83	0.84	0.82
Seq2seq [20]*	1.46	5.28	11.46	19.78	0.24	0.95	2.16	3.87	0.09	0.35	0.80	1.41	0.91	0.87	0.87	0.83
Seq2seq-SPL	1.57	5.00	10.01	16.43	0.27	0.94	2.01	3.45	0.10	0.36	0.79	1.36	0.91	0.87	0.87	0.84
Seq2seq-sampling [20]*	1.71	5.15	9.71	15.15	0.32	1.00	1.97	3.14	0.12	0.39	0.77	1.23	0.88	0.86	0.87	0.85
Seq2seq-sampling-SPL	1.71	5.13	9.60	14.86	0.31	0.97	1.91	3.04	0.12	0.38	0.74	1.18	0.89	0.86	0.88	0.85
Seq2seq-dropout	1.26	4.41	9.24	15.46	0.23	0.84	1.82	3.13	0.09	0.33	0.71	1.21	0.92	0.88	0.88	0.85
Seq2seq-dropout-SPL	**1.26**	4.26	8.67	14.23	0.23	0.81	1.74	2.96	0.09	0.32	0.68	1.16	0.92	0.89	0.89	0.86
QuaterNet [25]*	1.49	4.70	9.16	14.54	0.26	0.89	1.83	3.00	0.10	0.34	0.71	1.18	0.90	0.87	0.88	0.85
QuaterNet-SPL	1.34	4.25	8.39	13.43	0.25	0.83	1.71	2.83	0.09	0.32	0.67	1.10	0.91	0.88	0.89	0.86
RNN	1.69	5.23	10.18	16.29	0.31	1.05	2.17	3.62	0.12	0.41	0.85	1.43	0.89	0.85	0.86	0.83
RNN-SPL	1.33	**4.13**	**8.03**	**12.84**	**0.22**	**0.73**	**1.51**	**2.51**	**0.08**	**0.28**	**0.57**	**0.96**	**0.93**	**0.90**	**0.90**	**0.88**

Table 2: **AMASS results** of the base models with and without the proposed SP-layer. We report normalized area-under-the-curve (AUC) for PCK values (higher is better, maximum is 1). For the remaining metrics, lower is better. “Seq2seq” and “Seq2seq-dropout” are trained by using ground-truth inputs. "-dropout" applies 0.1 dropout on the inputs. “*” indicates our evaluation of this model. Note that models with SPL perform better except on short-term predictions for “Seq2seq” model.

Positional Following Pavllo *et al.*’s [25] suggestion, we introduce a positional metric. This metric simply performs forward kinematics on $\boldsymbol{x}_t$ and $\hat{\boldsymbol{x}}_t$ to obtain 3D joint positions $\boldsymbol{p}_t$ and $\hat{\boldsymbol{p}}_t$, respectively. It then computes the Euclidean distance per joint. We normalize the skeleton bones such that the right thigh bone has unit length.

$$L_{pos}(t) = \frac{1}{|\mathcal{X}_{test}|} \sum_{\boldsymbol{x}_t \in \mathcal{X}_{test}} \frac{1}{K} \sum_{k} \left\| \boldsymbol{p}_t^{(k)} - \hat{\boldsymbol{p}}_t^{(k)} \right\|_2 \qquad (9)$$

PCK In cases where large errors occur, the value of L_{pos} can be misleading. Hence, following the 3D (hand) pose estimation literature [13, 22, 28, 34], we introduce PCK by computing the percentage of predicted joints lying within a spherical threshold ρ around the target joint position, i.e.

$$PCK(\boldsymbol{x}_t, \hat{\boldsymbol{x}}_t, \rho) = \frac{1}{K} \sum_{k} \mathbb{I}\left[\left\| \boldsymbol{p}_t^{(k)} - \hat{\boldsymbol{p}}_t^{(k)} \right\|_2 \leq \rho \right]$$

$$L_{pck}(t, \rho) = \frac{1}{|\mathcal{X}_{test}|} \sum_{\boldsymbol{x}_t \in \mathcal{X}_{test}} PCK(\boldsymbol{x}_t, \hat{\boldsymbol{x}}_t, \rho) \qquad (10)$$

where $\mathbb{I}[\cdot]$ returns 1 if its input is true, and 0 otherwise. Note that for PCK we do not sum, but average, until time step t.

6.2. Results

Tab. 2 summarizes the performance of the three model variants, each with and without the SP-layer. We trained the base models with minimal modifications, i.e. design, training objective and regularizations are kept intact. We use angle-axis, quaternion and rotation matrix representations for Seq2seq, QuaterNet, and RNN models, respectively. To make a fair comparison, we run hyper-parameter search on the batch size, cell type, learning rate and hidden layer size.

Unlike on H3.6M, LSTM cells consistently outperform GRUs on AMASS for the Seq2seq and RNN models. Different from [20], we also train the Seq2seq model by applying dropout on the inputs similar to our RNN architecture. QuaterNet gives its best performance with GRU cells while some fine-tuning for the teacher forcing ratio is necessary.

In all settings, the Seq2seq models fail to give competitive performance on this large-scale task and are sometimes outperformed by the zero-velocity baseline proposed by Martinez *et al.* [20]. QuaterNet shows a strong performance and is in fact the closest vanilla model to the SPL variants. However, our SP-layer still improves the QuaterNet results further. The contribution of the SP-layer is best observable on the RNN model. With the help of a larger dataset, the proposed RNN-SPL achieves the best results under different metrics and prediction horizons. Fig. 4 compares two baseline methods for 400 millisecond predictions with their corresponding SPL extension for different choices of the threshold ρ. The RNN-SPL consistently outperforms other methods. More results are shown in the appendix Sec. 8.3.

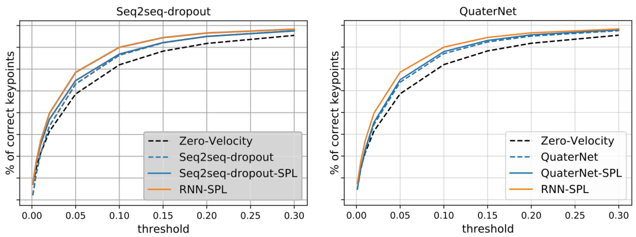

Figure 4: **PCK curves** of the best Seq2seq variant and QuaterNet with and without SPL on AMASS for 400 ms predictions. More results are shown in appendix Sec. 8.3.

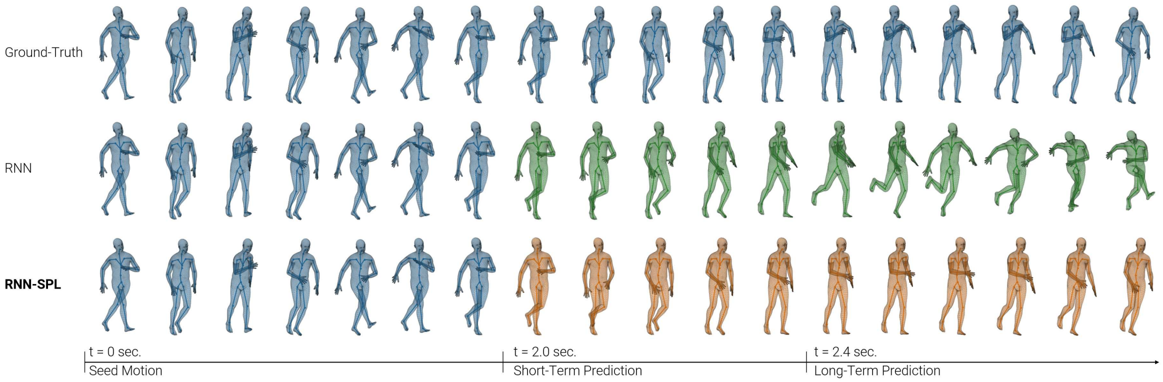

Figure 5: **Qualitative Comparison on AMASS.** We use a 2-second seed sequence and predict the next 1 second (60 frames). The last pose of the seed and the first pose of the prediction sequences are consecutive frames. Note that there is no transition problem. *Top:* Ground-truth sequence. *Middle:* Output of the vanilla RNN which quickly deteriorates. *Bottom:* The same RNN model augmented with our SP-layer. It produces accurate short-term predictions as well as natural long-term motion.

Please also note the complementary effect of the proposed metrics in Tab. 2. The Seq2seq-dropout-SPL model at 100 ms shows a significant improvement (1.26) w.r.t. the Euler angle metric, and in fact achieves the best result across all models. However, this is no longer the case when we look at the proposed metrics. The model performs marginally worse than the best performing model, RNN-SPL, in these metrics. The joints closer to the root of the kinematic chain have a much larger impact on the overall pose since wrong rotations propagate to all the child joints on the chain. This effect might be ignored when only local rotations are considered, which is the case for L_{eul}. L_{angle} and L_{pos} account for this by first unrolling the kinematic chain.

In line with [25, 33], we report that the residual connection from [20] is very effective for short-term predictions. All models we trained performed better with the residual connection irrespective of the dataset or pose representation.

6.3. Ablation Study

To study the SPL in more depth we conduct an ablation study presented in Tab. 3. We observe that the main performance boost is achieved by the decomposition of the output layer and the per-joint loss in Eq. (6). While the per-joint-loss alone (i.e., without SPL) is not beneficial on H3.6M, on AMASS its application alone already helps (*RNN-per-joint*). Assuming independent joints without modelling any hierarchy (*RNN-SPL-indep.*) improves the results further. Introducing hierarchy into the prediction layer either in reverse or random order performs often similar or better. However, introducing the spatial dependencies according to the kinematic chain (*RNN-SPL*) yields the best results with the exception of the positional metric.

	AMASS			H3.6M
	Euler	Joint Angle	Pos.	Walking
RNN	16.44	3.570	1.396	0.900
RNN-per-joint	13.13	2.573	0.986	0.950
RNN-SPL-indep.	12.96	2.552	0.982	0.836
RNN-SPL-random	12.98	2.547	0.980	0.863
RNN-SPL-reverse	13.03	2.543	**0.973**	0.849
RNN-SPL	**12.85**	**2.533**	0.975	**0.772**

Table 3: **Ablation study** on AMASS and H3.6M (*walking*) for 400 ms predictions. Each entry is an average over 5 randomly initialized trainings. Please refer to Sec. 6.3 for detailed explanations and the appendix for more results.

7. Conclusion

We introduce prior knowledge about the human skeletal structure into a neural network by means of a structured prediction layer (SPL). The SP-layer explicitly decomposes the pose into individual joints and can be interfaced with a variety of baseline architectures. We furthermore introduce AMASS, a large-scale motion dataset, and several metrics to the task of motion prediction. On AMASS, we empirically show that for any baseline model, any metric, and any input representation, it is better to use the proposed SP-layer. The simple RNN model augmented with the SP-layer achieved state-of-the-art performance on the new AMASS benchmark.

Acknowledgements We thank the reviewers for their insightful comments and Martin Blapp for fruitful discussions. This work was supported in part by the ERC Grant OPTINT (StG-2016-717054). We thank the NVIDIA Corporation for the donation of GPUs used in this work.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Judith Bütepage, Michael J. Black, Danica Kragic, and Hedvig Kjellström. Deep representation learning for human motion prediction and classification. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1591–1599, 2017.

[3] Judith Bütepage, Hedvig Kjellström, and Danica Kragic. Anticipating many futures: Online human motion prediction and generation for human-robot interaction. In *2018 IEEE International Conference on Robotics and Automation, ICRA 2018, Brisbane, Australia, May 21-25, 2018*, pages 1–9, 2018.

[4] Kyunghyun Cho, Bart van Merriënboer, Çağlar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua 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)*, pages 1724–1734, Doha, Qatar, Oct. 2014. Association for Computational Linguistics.

[5] Fernando De la Torre, Jessica Hodgins, Adam Bargteil, Xavier Martin, Justin Macey, Alex Collado, and Pep Beltran. Guide to the carnegie mellon university multimodal activity (cmu-mmac) database. *Robotics Institute*, page 135, 2008.

[6] Xiaoxiao Du, Ram Vasudevan, and Matthew Johnson-Roberson. Bio-lstm: A biomechanically inspired recurrent neural network for 3d pedestrian pose and gait prediction. *IEEE Robotics and Automation Letters (RA-L)*, 2019. accepted.

[7] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 4346–4354, Washington, DC, USA, 2015. IEEE Computer Society.

[8] Partha Ghosh, Jie Song, Emre Aksan, and Otmar Hilliges. Learning human motion models for long-term predictions. In *2017 International Conference on 3D Vision, 3DV 2017, Qingdao, China, October 10-12, 2017*, pages 458–466, 2017.

[9] Daniel Holden, Jun Saito, and Taku Komura. A deep learning framework for character motion synthesis and editing. *ACM Trans. Graph.*, 35(4):138:1–138:11, July 2016.

[10] Daniel Holden, Jun Saito, Taku Komura, and Thomas Joyce. Learning motion manifolds with convolutional autoencoders. In *SIGGRAPH Asia 2015 Technical Briefs*, SA '15, pages 18:1–18:4, New York, NY, USA, 2015. ACM.

[11] Yinghao Huang, Manuel Kaufmann, Emre Aksan, Michael J. Black, Otmar Hilliges, and Gerard Pons-Moll. Deep inertial poser: Learning to reconstruct human pose from sparse inertial measurements in real time. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 37:185:1–185:15, Nov. 2018.

[12] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, jul 2014.

[13] Umar Iqbal, Pavlo Molchanov, Thomas Breuel, Juergen Gall, and Jan Kautz. Hand pose estimation via latent 2.5d heatmap regression. In *ECCV (11)*, volume 11215 of *Lecture Notes in Computer Science*, pages 125–143. Springer, 2018.

[14] Ashesh Jain, Amir Roshan Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 5308–5317, 2016.

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[16] Kyoungoh Lee, Inwoong Lee, and Sanghoon Lee. Propagating LSTM: 3d pose estimation based on joint interdependency. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VII*, pages 123–141, 2018.

[17] Sijin Li, Weichen Zhang, and Antoni B. Chan. Maximum-margin structured learning with deep networks for 3d human pose estimation. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 2848–2856, 2015.

[18] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM Transactions on Graphics (TOG)*, 34(6):248, 2015.

[19] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. Amass: Archive of motion capture as surface shapes. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2019.

[20] Julieta Martinez, Michael J. Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017*, Piscataway, NJ, USA, July 2017. IEEE.

[21] Francesc Moreno-Noguer. 3d human pose estimation from a single image via distance matrix regression. In *CVPR*, pages 1561–1570. IEEE Computer Society, 2017.

[22] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[23] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[24] Dario Pavllo, Christoph Feichtenhofer, Michael Auli, and David Grangier. Modeling human motion with quaternion-based neural networks. *CoRR*, abs/1901.07677, 2019.

[25] Dario Pavllo, David Grangier, and Michael Auli. Quaternet: A quaternion-based recurrent model for human motion. In *British Machine Vision Conference 2018, BMVC 2018, Northumbria University, Newcastle, UK, September 3-6, 2018*, page 299, 2018.

[26] L. Sigal, A.O. Balan, and M.J. Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal on Computer Vision (IJCV)*, 87(1):4–27, 2010.

[27] Gregory G. Slabaugh. Computing euler angles from a rotation matrix. http://www.gregslabaugh.net/publications/euler.pdf, last accessed 21.03.2019.

[28] Adrian Spurr, Jie Song, Seonwook Park, and Otmar Hilliges. Cross-modal deep variational hand pose estimation. In *CVPR*, 2018.

[29] Xiao Sun, Jiaxiang Shang, Shuang Liang, and Yichen Wei. Compositional human pose regression. In *ICCV*, pages 2621–2630. IEEE Computer Society, 2017.

[30] Graham W. Taylor, Geoffrey E. Hinton, and Sam T. Roweis. Two distributed-state models for generating high-dimensional time series. *Journal of Machine Learning Research*, 12:1025–1068, 2011.

[31] Bugra Tekin, Isinsu Katircioglu, Mathieu Salzmann, Vincent Lepetit, and Pascal Fua. Structured prediction of 3d human pose with deep neural networks. In *Proceedings of the British Machine Vision Conference 2016, BMVC 2016, York, UK, September 19-22, 2016*, 2016.

[32] Timo von Marcard, Bodo Rosenhahn, Michael Black, and Gerard Pons-Moll. Sparse inertial poser: Automatic 3d human pose estimation from sparse imus. *Computer Graphics Forum 36(2), Proceedings of the 38th Annual Conference of the European Association for Computer Graphics (Eurographics)*, pages 349–360, 2017.

[33] Yuxiong Wang, Liang-Yan Gui, Xiaodan Liang, and Jose M. F. Moura. Adversarial geometry-aware human motion prediction. In *European Conference on Computer Vision (ECCV)*. Springer, October 2018.

[34] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. https://arxiv.org/abs/1705.01389.