Human Motion Prediction via Learning Local Structure Representations and Temporal Dependencies

Xiao Guo, Jongmoo Choi

University of Southern California, Department of Computer Science
xiaoguo@usc.edu, jongmooc@usc.edu

Abstract

Human motion prediction from motion capture data is a classical problem in the computer vision, and conventional methods take the holistic human body as input. These methods ignore the fact that, in various human activities, different body components (limbs and the torso) have distinctive characteristics in terms of the moving pattern. In this paper, we argue local representations on different body components should be learned separately and, based on such idea, propose a network, Skeleton Network (SkelNet), for long-term human motion prediction. Specifically, at each time-step, local structure representations of input (human body) are obtained via SkelNet's branches of component-specific layers, then the shared layer uses local spatial representations to predict the future human pose. Our SkelNet is the first to use local structure representations for predicting the human motion. Then, for short-term human motion prediction, we propose the second network, named as Skeleton Temporal Network (Skel-TNet). Skel-TNet consists of three components: SkelNet and a Recurrent Neural Network, they have advantages in learning spatial and temporal dependencies for predicting human motion, respectively; a feed-forward network that outputs the final estimation. Our methods achieve promising results on the Human3.6M dataset and the CMU motion capture dataset, and the code is publicly available [1].

Introduction

Human motion prediction from motion capture (mocap) data is attracting a significant attention in recent years, which has been applied in a variety of fields: human pose tracking (Taylor et al. 2010; Tekin et al. 2016), Physics-Based Motion synthesis (Liu, Hertzmann, and Popović 2005; Arikan, Forsyth, and O'Brien 2003) and human-computer interaction (Koppula and Saxena 2016). However, estimating physical parameters, like skeleton joint angles, used to represent human poses from mocap data is still a challenging issue, as the realistic human motion is involved with complexities in kinematics and variations of the dynamic pattern, which are difficult to predict.

Our goal is to forecast future human poses from mocap data via proposed networks, in which *each mocap feature vector is a set of skeleton joint angles of the human pose*. In

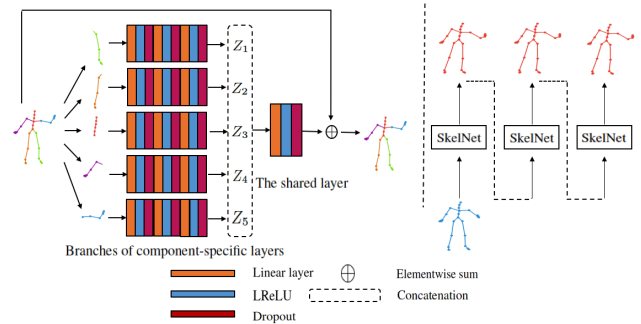[1]https://github.com/CHELSEA234/SkelNet_motion_prediction

Figure 1: *Left*: An illustrative sketch of proposed SkelNet. The human pose is divided into five non-overlapping parts depicted in different colors, which are fed accordingly into five branches of component-specific layers for learning local structure representations. z_1, z_2, z_3, z_4, z_5 respectively denote representations for different body components. The shared layer uses representations of local structure to predict the future human pose. *Right*: During both training and testing, SkelNet is fed with one seed mocap vector, then predicts the mocap sequence by always sampling its own generated samples.

fact, our prediction task can be divided into two subtasks: long-term and short-term predictions. In long-term scenario, errors can accumulate severely with long time horizons; in short-term, motions are more certain and constrained by the temporal coherence. Many previous methods (Taylor, Hinton, and Roweis 2007; Taylor et al. 2010; Fragkiadaki et al. 2015; Jain et al. 2016; Martinez, Black, and Romero 2017; Butepage et al. 2017) have achieved promising results on these two subtasks by using deep learning approaches. However, we observe that different physical components (limbs and the torso) of the human body participate into actions to varying degrees. These components have different moving dynamics and should be processed separately, but the previous work is fed with the human pose that is never fully partitioned in a component-specific way, this violates the kinematics principle of the human body, and may bring in intra-pose interventions in generating future human poses.

In this paper, we propose a novel network at first for long-term human motion prediction, named as Skeleton Net-

work (SkelNet). Unlike the previous work ignoring distinctive characteristics in terms of the moving pattern of different human body components, SkelNet uses branches of component-specific layers for learning local structure representations of different human body components. Our Skel-Net (Figure 1) is a variant to the standard feed-forward network, after being fed with the seed mocap feature vector(s), it generates a mocap sequence by always sampling previous self-generated samples. Specifically, instead of taking the holistic human pose as input, we divide it into five non-overlapping parts according to the human physical structure, and then separately feed these five parts into Skel-Net's five branches of component-specific layers. SkelNet's component-specific layers learn local spatial representations of input at the current time-step, and a shared layer is responsible to use these local representations for estimating the mocap feature vector at the next time-step. To the best of our knowledge, SkelNet is the first to predict the human motion via local spatial representations of the human pose, where its branch structure learns component-independent information from different human body-part domains. Moreover, SkelNet also derives its generalization capability from a special network design, which is based on a set of simple ideas, such as adding the residual connection, using dropout and the nonlinear activation function.

Furthermore, temporal dependencies interpret temporal coherence imposed by the human motion over time horizons, which is important for predicting future human poses, especially when motion is more certain within a short time range. Hence, we propose the second network, Skeleton Temporal Network (Skel-TNet), for short-term prediction task, it employs SkelNet in the company of Recurrent Neural Network (RNN) that is known as an effective technique for modeling temporal dynamics. In fact, our Skel-TNet has three components (Figure 3), aside from SkelNet, we use a RNN with one Gate Recurrent Unit (GRU) (Cho et al. 2014), named as C-RNN, which can efficiently learn temporal dynamics of the human pose sequence. The third component is Merging Network, an another feed-forward network that aims at generating the final estimated mocap sequence. Skel-TNet is a multistage processing framework, three components need to be trained separately yet an end-to-end fashion is maintained in the evaluation. It demonstrates a new way for predicting short-term human motion: to leverage results generated from pre-trained models, which respectively have advantages in learning local spatial structure representations and temporal dependencies.

In summary, our contributions are: 1) SkelNet, a new network that uses local structure representations for predicting long-term human motion. This is the first time that local spatial representations are used for human motion prediction, 2) Skel-TNet, a new network for predicting short-term human motion, it takes advantages of capabilities on learning local structure representations and temporal dependencies from proposed SkelNet and a GRU-based RNN, respectively, 3) experimental results demonstrate that our methods achieve promising results on human motion prediction, compared to the state-of-art method, and SkelNet exhibits the meaningful robustness towards noise.

Related work

Human motion prediction in mocap data. Many works are based on traditional statistical methods for predicting or synthesizing the human motion (L'opez-M'endez et al. 2012; Zhao and Ji 2018; Wang, Fleet, and Hertzmann 2005; Wu and Shao 2014). Meanwhile, deep learning approaches have also achieved remarkable accomplishments. Specifically, (Fragkiadaki et al. 2015) present two end-to-end discriminatively trained models: Encoder-Recurrent-Decoder (ERD) and a LSTM recurrent neural network (LSTM-3LR), for modeling human kinematics from videos and mocap data. In (Jain et al. 2016), authors cast an arbitrary, scalable and jointly trainable stacked RNN based LSTMs, called structural RNN (SRNN). Then residual RNN (RRNN) is proposed by (Martinez, Black, and Romero 2017), which solve issue of discontinuity in initial frames by a relatively simple and intuitive way, and after that, (Ghosh et al. 2017) build model for predicting human motion in the long run. Most recently, (Li et al. 2018) use a convolutional sequence to sequence (seq2seq) (Sutskever, Vinyals, and Le 2014) model (CSS) for learning both global spatial dependencies and long-term temporal dependencies. It is noteworthy that one common feature of some existing methods including (Butepage et al. 2017) is their heavy reliance on long prior information encoded from many previous observed mocap frames, in contrast, our SkelNet does not need such help, still being able to produce encouraging results.

Architecture with Multi-stage processing. Several previous methods construct frameworks which are multi-stage processing, or combine multiple methods that are streamlined processing, for obtaining promising results on various tasks: object detection and segmentation (Girshick et al. 2014); face hallucination (Yu and Porikli 2017); various tasks in image synthesis (Huang et al. 2017; Zhang et al. 2017). In short, networks operating in multi-stages can benefit from being conditioned on intermediate results, as well as specific capabilities provided by pretrained models. What is more, spatial and temporal dependencies are two ideas that have been applied in a variety of works relating to the human analysis: human motion tracking (Xu et al. 2018), 3D pose estimation (Fang et al. 2018), activity recognition (Liu et al. 2017b; 2017a) and robotics (Koppula and Saxena 2016), etc. Our Skel-TNet uses both local spatial dependencies and temporal dependencies for predicting the human motion, this is different from the previous work who (Jain et al. 2016) attempts to learn spatial and temporal dependencies altogether in one single largely complicated spatio-temporal graph, and who (Ghosh et al. 2017) obtains the mediocre result by training two independent networks jointly.

Proposed method

SkelNet

Architecture: As previously stated, many previous methods rely on using large number of observed frames as the prior information, but observed frames can be unavailable and corrupted with noise, so we choose a standard feed-forward

network as the the basic sequence generator (baseline) instead of the seq2seq model, it outputs the mocap vector $y_t(t \geq 1)$, as estimation to the one-step-ahead ground truth vector x_t, given self-generated mocap vectors (including seed one) so far (Figure 1). The residual connection (He et al. 2016) allows the network to predict the motion velocity instead of the whole human pose, this achieves encouraging results without taking considerable efforts, as demonstrated in (Martinez, Black, and Romero 2017). We also incorporate leaky Rectified Linear Units (LReLUs) (Maas 2013), dropout (Srivastava et al. 2014) into the network design for adding non-linearity and combating overfitting. Leveraging these simple ideas in the deep learning enhances the generalization capability of our baseline on human motion prediction, details are reported in Table 3.

The first several layers of SkelNet is divided into five branches. Being different with the previous work, regrading input data at each time-step as a complete one, we divide such input into five non-overlapping parts, according to the upper left/ right (arm), the lower left/ right (leg) and the torso. These five inputs are fed into five component-specific branches accordingly, and branches learn local structure representations of the human pose. Such partitioned data altogether with the branch structure make SkelNet have the higher generalization capability than that of networks used in the previous work. This is due to the fact that different physical parts of the human pose have different level involvements in certain activities, such that five different kinematical information of the human pose should be treated and processed separately. For example, activities like "walking" and "running" exhibit a large amount of variations in lower skeletal parts (two legs), yet upper parts (arms) matter less; people wave, lift and stretch out their arms to do "directions", "washing window" and "eating" while being probably hold legs and torso still.

The input layer in a conventional feed-forward network takes in data without partitioning and project the high dimensional vector as output, each dimension (value) is decided jointly by the spatial information of the entire human pose, namely five assigned groups altogether. This suggests that, throughout the entire neural network, every single unit considers global information of the complete human pose, this can cause intra-pose interventions, which hinder us from learning local structural representations of the body configuration for forecasting future poses. Conversely, in SkelNet, data from different groups are passed through branches accordingly, such simple modification largely overcomes downside of intra-pose interventions, focusing information of the locality like limb-specific or torso-specific information. Figure 2 demonstrates our idea, and more comprehensive results are reported in ablative study.

Robustness analysis: If input data is corrupted by noise, we conjecture that the previous work will witness larger declines in the prediction accuracy than SkelNet. Evidently, it is because that some of the previous work (e.g., RRNN and CSS) uses long prior information encoded from a large number of previous mocap frames, whose guidance meaning will
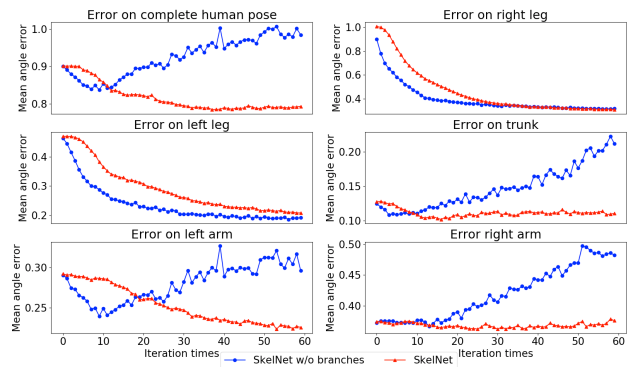

Figure 2: Six plots record Euler angle errors on the complete human pose and their five different human component groups, respectively. On predicting long-term "Eating" activity in the Human3.6M dataset, Red lines indicate errors from the SkelNet and Blue lines indicate error from the SkelNet without branches.

diminish if there are missing frames or skeleton poses corrupted by noise. On the contrary, SkelNet counts generalization capability on the particular network design, without the need of long prior information, so SkelNet is likely to make the more robust choice. However, this is partially proved. We evaluate models for the prediction task, where input is under the Gaussian noise. SkelNet outperforms previous methods on the CMU motion capture dataset but keeps comparable in the Human3.6M dataset, relating details will be reported in Table 4. Even so, SkelNet still exhibits meaningful robustness towards noise.

In short-term prediction, motion is more constrained by the temporal coherence, this has been demonstrated in (Fragkiadaki et al. 2015; Martinez, Black, and Romero 2017) and thereby the RNN-based method is a popular approach. However, our SKelNet constructed as a lightweight architecture cannot learn temporal dynamics as effectively as RNN-based methods equipped with LSTMs and GRUs, which are known as efficient ways for modeling temporal dependencies by adopting multiple gates. To handle this issue, we design Skeleton Temporal Network (Skel-TNet) based on SkelNet.

Skel-TNet

Architecture: For short-term human motion prediction, we propose Skel-TNet for predicting the human motion, in which SkelNet works with a RNN. As a matter of fact, the major motivation behind Skel-TNet is making use of different models' advantages in learning spatial and temporal dependencies, both of which play significant roles in our prediction task.

Skel-TNet has three components (Figure 3). Aside from SkelNet that predicts the human motion via learning local structure representations, we use a RNN based on GRU that can efficiently model temporal dynamics, we name it as C-RNN. C-RNN has one standard GRU, as a computationally less-expensive alternative to LSTMs, followed by two linear
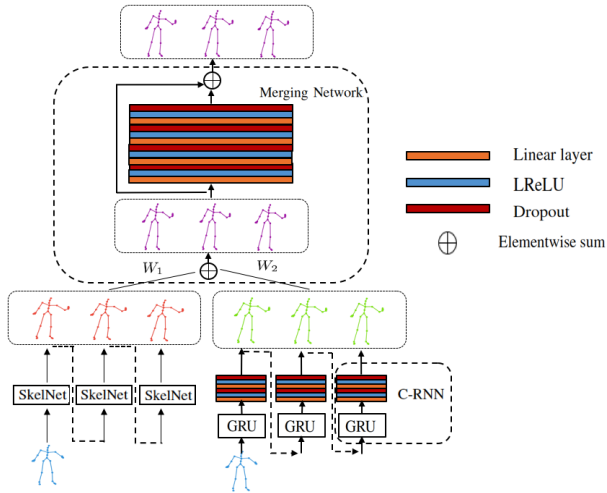
Figure 3: Skel-TNet has three main components: SkelNet, C-RNN and Merging Network. We train SkelNet and C-RNN at the first stage. Pre-trained models generate two sequences (Red and Green) as inputs to Merging Network, which outputs the final generated sequence (Purple). In testing, three components are maintained in an end-to-end fashion. At each time-step, the residual connection used to bridge input and output in C-RNN are not depicted.

layers with LReLUs, dropout and the residual connection. It predicts the future sequence given the seed mocap frame(s) along with self-geneated samples. Our C-RNN achieves more promising results via being trained by our proposed *Converging_loss*, which will be introduced later. SkelNet and C-RNN separately output two estimated sequences, as intermediate results, sent to the third component, which is an another feed-forward neural network, called Merging Network. Our Merging Network is responsible for the final estimated sequence and it has two trainable weights, which adaptively control how much of two intermediate results are used for generating the final prediction.

Learning: Optimizing RNNs by always feeding them the previous self-generated sample as input, this idea is firstly proposed by (Martinez, Black, and Romero 2017) on human motion prediction and called *Sampling-based loss*. Although such training strategy improves network's ability to recover from its own mistakes along the generation, we find that it often leads to the poor human-motion prediction performance, which is because the divergence exists in training between conditioned context (the self-generated sample) and the ground truth. How to reduce the negative impact brought by this type of divergence in training, while keep updating RNNs' ability to recover from its own mistakes is the issue we need to tackle. Hence, we design *Converging loss* ($\mathcal{L}_{conv}$), which is composed of two terms: *Positive loss* ($\mathcal{L}_{pos}$) and *Negative loss* ($\mathcal{L}_{neg}$), and α, β is the weight associated with $\mathcal{L}_{pos}$ and $\mathcal{L}_{neg}$.

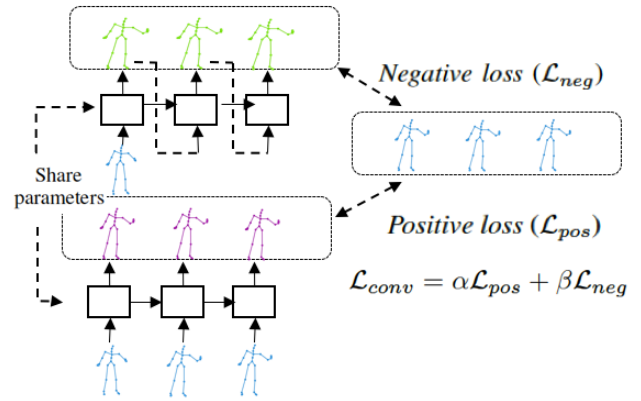$$\mathcal{L}_{conv} = \alpha\mathcal{L}_{pos} + \beta\mathcal{L}_{neg} \qquad (1)$$

Figure 4: The first output sequence (Green) is generated from the RNNs model conditioned on its own output at previous time-steps and the seed mocap frame(s); The second output sequence (Purple) is generated from the RNNs model conditioned on the ground truth at all time-steps; $\mathcal{L}_{pos}$ and $\mathcal{L}_{neg}$ is computed between them (Green and Purple sequence) and the ground truth sequence (Blue), respectively.

Specifically, these two terms are Euclidean distances between the ground truth sequence and the generated sequence, but the generated sequence used in $\mathcal{L}_{pos}$ is from RNNs always being fed by the ground truth and sequence used in $\mathcal{L}_{neg}$ is retrieved from RNNs constantly taking in self-generated samples, as shown in Figure 4. $\mathcal{L}_{conv}$ encourages RNNs to generate sequence conditioned on the ground truth in training via $\mathcal{L}_{pos}$, meanwhile forces RNNs to keep learning how to correct from its own mistakes via $\mathcal{L}_{neg}$. We empirically find that if we replace *Sampling-based loss* with $\mathcal{L}_{conv}$ as the objective function in training, improvements of prediction performance on RRNN and C-RNN are evident, so we choose to train C-RNN via $\mathcal{L}_{conv}$. After optimizing SkelNet and C-RNN, we use these two pertained models to generate two mocap sequences, which are sent to Merging Network that is trained by minimizing the Euclidean distance between its estimation and the ground truth. Although three components need to be trained independently, Skel-TNet maintains the end-to-end fashion in the testing phase.

Understanding: For $\mathcal{L}_{conv}$, we in fact instantiate a longstanding idea on the optimization for generative RNNs: monitoring behaviors of RNNs in the "free-running" (negative phase) mode as well as the "teacher forcing" (positive phase) mode (Williams and Zipser 1989). Recently, one of famous instances of this idea turns out to be the professor-forcing algorithm (Lamb et al. 2016), nevertheless **such idea has yet to be applied on the human modeling community until our** $\mathcal{L}_{conv}$. Furthermore, $\mathcal{L}_{conv}$ serves as a way of complementing information of the ground truth in the decoding phase. Moreover, chances are that $\mathcal{L}_{conv}$ could be deployed into other generation problems, such as hand-writing images reconstruction, speech generation, etc. Furthermore, we need to claim three points for the more

solid explanation to the network design of Skel-TNet. Firstly, employing both SkelNet and C-RNN is for learning both structure representations and temporal dependencies, not for small gainings in performance, as results can be further enhanced by using the deeper feed-forward network or stacking multiple GRUs. Secondly, the main purpose of our Merging Network is designed to combine and balance two intermediate results into single one as the final output, its predictive ability is limited but which is what (Butepage et al. 2017) mainly focuses on. This difference in motivations separates our Merging Network from their proposed network, even though it is true that these two share similar structures. Thirdly, our Skel-TNet is the first method that employs a multi-stage processing network, and each component is conceptually simple.

In summary, Skel-TNet instantiates the idea that human motion prediction can be achieved by a multi-staged processing network, which makes use of different model's advantages, namely, SkelNet can learn local spatial structure effectively, C-RNN can learn temporal dynamics effectively.

Experiment

In this section, we introduce two datasets and implementation details. Then we evaluate our proposed models for several different tasks in human motion prediction. Our experiments include comparison with many previous methods, including ERD, LSTM-3LR(Fragkiadaki et al. 2015), SRNN(Jain et al. 2016), RNN_D (Lin and Amer 2018), RRNN(Martinez, Black, and Romero 2017) and CSS(Li et al. 2018). In the end, ablative experiments are carried out for showing each component's contribution.

Experiment setup

Dataset and preprocessing: Two widely used public benchmarks are chosen in our experiments: 1) Human3.6M (Ionescu et al. 2014) : It has 15 different activity categories performed by professional actors (subjects) from ordinary life: "walking", "eating", "smoking", etc. It is the largest human motion capture dataset and also has been winning increasing popularity recently. Following the previous method(Martinez, Black, and Romero 2017), we use the sequences given by the subject five for testing, while rest sequences for training. The detail can be found in the public[2], 2) CMU motion capture (CMU mocap) dataset (Lab): it contains 2235 recordings belonging to five major activity categories, being performed by 144 different subjects. This is a challenging dataset as it contains more complex activities, like "basketball", "soccer". Data is recorded with the mocap system and a pose are represented with 38 joints in 3D space. As with (Li et al. 2018), eight actions are selected for our experiments, who are already divided into as two parts for training and testing, the code is publicly available[3] . Also, two datasets are preprocessed in the same way as

---

[2]https://github.com/una-dinosauria/human-motion-prediction

[3]https://github.com/chaneyddtt/Convolutional-Sequence-to-Sequence-Model-for-Human-Dynamics/tree/master/src

the previous work, in which still joints are removed and Euler angles are converted to exponential map representation, making human skeleton pose invariant to the orientation and avoid the gimbal lock effect.

Implementation: We implement our methods using the Tensorflow as the backend. Regarding SkelNet, each branch consists of three linear layers, dimensions are 64, 128, 64. The last layer is of the same dimension as input, namely dimension of each time-step's mocap frame (54 and 70 dimensions for data in two datasets, respectively), denoted as *input dimension*. We adopt the Gradient Descent optimizer with learning rate 0.01 [4]. In Skel-TNet, C-RNN has a GRU with 1024 units, followed by two fully-connected layers (512, *input dimension*), this network is trained by $\mathcal{L}_{conv}$, where weights ahead of $\mathcal{L}_{pos}$, $\mathcal{L}_{neg}$ are set as 1 and 0.1. The learning rate is set as 5e-5 for Skel-TNet, with the gradient descent optimizer. Merging Network is as depicted in Figure 3, dimensions are 1024, 512, 512, *input dimension*, and two input weights are initialized as 0.5. We use the adam optimizer with learning rate 0.01 to train Merging Network. Throughout the entire network, LReLUs are with the negative slope as 0.2 and dropouts are with rate 0.2 to dropout.

Metric: We adhere to the evaluation metric in the previous work: we measure Euclidean distances between our predictions and the ground truth in the angle-space for increasing time horizons, then compute the average error at each frame over eight randomly sampled test sequences. The result is reported in Table 2 that has three commonly used representative actions. The random seed that generates eight sequences is fixed as the same one in the previous work for the fair comparison. Also, we report mean errors over all frames along eight randomly chosen test sequences for each actions in two datasets, the result is denoted as the MoF (mean error of frames) for simplicity.

Results and comparisons

Long-term motion prediction: SkelNet is trained by minimizing the prediction error over 1000 ms in the future, and we compare it with recently proposed methods (CSS and RRNN). We report results in Table 1(a), whose rightmost column demonstrates that our results catch up with or even surpass the state-of-the-art method. We notice that RRNN suffers from a severe overfitting issue and relatively poor prediction performances (e.g., MoF in predicting "Walking", "Eating" activities), even though it has been considered as the flagship work previously. CSS simply takes a large number of complete human poses into the network, which we believe cannot learn spatial dependencies imposed by each human skeleton pose, on the contrary, SkelNet explicitly uses branches of component-specific layers to learn the spatial dependencies. Except "Walking-dog" activity in the Human3.6M dataset, SkelNet outperforms two competitors in all other scenarios, and the enhancement on predicting "BasketballSignal" activity is remarkable. Qualitatively, we are able to see clear movements in different motion sequences generated by SkelNet, results keep physically

---

[4]The choice is to make the magnitude of $\mathcal{L}_{pos}$ and $\mathcal{L}_{neg}$ are of same scale.

							Human3.6M									
	Walk	Eat	Somke	Discuss	Direct	Greet	Phone	Pose	Purch	Sitting	SittingD	Photo	Wait	WalkD	WalkT	Average/Standard Deviation
RRNN	0.97	1.10	1.30	1.33	1.60	1.98	1.67	1.89	1.71	1.52	1.90	1.31	1.60	1.70	1.07	1.51/**0.32**
CSS	0.71	0.78	0.99	1.27	1.00	1.47	1.53	1.75	1.44	1.21	1.33	0.91	1.55	**1.45**	0.83	1.22/0.32
SKelNet	**0.69**	**0.77**	0.96	**1.21**	**0.96**	**1.46**	**1.50**	**1.56**	**1.41**	**1.14**	**1.24**	**0.84**	1.57	1.52	**0.78**	**1.17/0.32**
SKel-TNet	0.70	**0.77**	**0.95**	1.22	**0.96**	1.47	1.52	1.58	1.44	1.17	1.26	0.85	**1.51**	1.52	0.80	1.18/**0.32**

					CMU mocap dataset				
	Walk	Run	DirectTraffic	Soccer	Basketball	WashWindow	Jump	BasketballSignal	Average/Standard Deviation
CSS	0.56	**0.48**	1.04	0.79	1.54	0.86	1.55	0.65	0.93/**0.42**
SKelNet	**0.51**	0.57	**0.99**	**0.74**	**1.50**	**0.79**	**1.54**	**0.41**	**0.88**/0.43
SKel-TNet	0.53	0.59	**0.99**	0.76	1.52	0.81	1.59	0.43	0.90/0.44

(a) Long-term human motion prediction.

							Human3.6M									
	Walk	Eat	Somke	Discuss	Direct	Greet	Phone	Pose	Purch	Sitting	SittingD	Photo	Wait	WalkD	WalkT	Average/Standard Deviation
RRNN	0.51	0.61	0.77	0.91	1.06	1.20	1.28	1.31	1.08	0.94	1.11	0.82	1.02	1.04	0.75	0.97/0.23
CSS	0.49	0.44	0.63	0.73	0.65	**1.00**	**1.17**	0.95	0.92	0.76	0.83	0.57	0.82	0.99	0.53	0.77/**0.21**
SKelNet	0.49	0.46	**0.60**	**0.70**	0.62	1.03	1.21	0.77	0.89	0.76	**0.80**	**0.56**	0.85	1.00	0.51	0.76/**0.21**
SKel-TNet	**0.48**	**0.41**	0.61	**0.70**	**0.62**	**1.00**	1.19	**0.76**	**0.86**	**0.75**	**0.80**	0.57	**0.81**	**0.96**	**0.50**	**0.73/0.21**

					CMU mocap dataset				
	Walk	Run	DirectTraffic	Soccer	Basketball	WashWindow	Jump	BasketballSignal	Average/Standard Deviation
CSS	0.41	**0.46**	0.60	0.63	0.86	0.51	**0.90**	0.58	0.61/**0.18**
SKelNet	**0.34**	0.56	0.57	0.55	0.84	0.55	0.96	0.38	0.60/0.21
SKel-TNet	0.36	0.52	**0.54**	**0.48**	**0.81**	**0.51**	**0.90**	**0.32**	**0.55**/0.21

(b) Short-term human motion prediction.

Table 1: Long-term and short-term human motion prediction errors reported in MoF on each activity of the Human3.6M dataset and the CMU mocap dataset. Bold means the best performance.
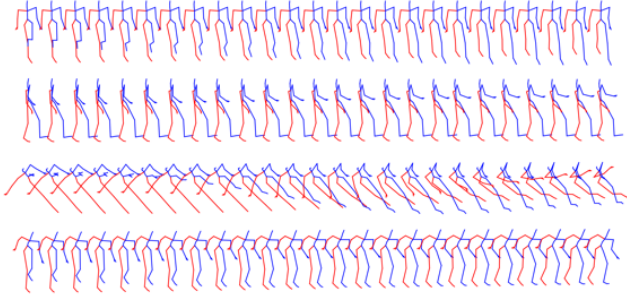

Figure 5: Qualitative results on "Walking", "DirectTraffic", "Soccer" and "Basketball" activities (from top to down) in the CMU mocap dataset.

meaningful and realistic. In "Walking" and "Basketball", stick man is with footstep changed; In "DirecTraffic", the left hand is raised; In "Soccer", stick man kicks ball out by different body positions.

In order to have the more concrete analysis on proposed networks' prediction performance, we also train Skel-TNet on long-term prediction task. As a result, Skel-TNet performs slightly better than CSS regarding the averaging MoF, which means it also can handle long-term prediction task.

Shot-term motion prediction: In this task, we firstly train Skel-TNet by minimzing the prediction error over future 400 ms. We report results measured by MoF for every single activities in two datasets in Table 1(b). We can notice that Skel-TNet surpasses or at least catches up with CSS, specifically, results are more accurate: lowering averaging errors of the second best results on two datasets by 5% and 10%, respectively. Although SkelNet is designed for long-term prediction task, we also train it for short-term prediction task, and it performs comparably with the state-of-the-art method on

many activity predictions, and it achieves slightly lower averaging MoF than CSS.

In Table 2, we compare with multiple networks for human motion prediction on several time-steps (i.e. 80, 160, 240, 320 ms). We can notice that the best result constantly fallen into one of our methods. Albeit slight increasing margins in Table 2, we still believe that these margins can be enlarged by extending Skel-TNet's components to more complicated ones, which are with higher learning capability regarding each spatial and temporal dependencies.

Ablative study

SkelNet: We firstly study the effectiveness of the baseline, a standard feed-forward network for predicting human motion over long-term, by removing LReLUs, dropout and branches (w/o branches w/o LReLUs w/o dropout), this cause the error to increase; its performance can be improved by adding the dropout (w/o branches w/o LReLUs) or adding both the dropout and LReLUs (w/o branches). Furthermore, we illustrate the difference between full SkelNet (full) with or without branches (w/o branches), the difference indicates that branches in SkelNet do improve prediction performance, we think this is because intra-pose interventions can be largely tackled. This idea is further supported by Figure 2, except Eular angle error on left arm, errors on other components and complete human pose are reduced. Moreover, our Skel-Net's effectiveness will not be restricted when long prior is available, as shown by the first two rows.

Furthermore, we compare with two networks with 3 branches of layers, using two different groupings: left (leg+arm), torso, right(leg+arm), denoted as Skel_LR; (left+right) arms, torso, (left+right) legs, denoted as Skel_UD; Performance of Skel_LR and Skel_UD are both worse than original SkelNet with five branches of layers that better capture component-specific information.

Also, we investigate the robustness of SkelNet. We train SkelNet on long-term prediction, input data in both training

millisecond	Eat 80	160	240	320	400	Smoke 80	160	240	320	400	Discuss 80	160	240	320	400
ERD(ICCV'15)	1.27	1.45		1.66	1.80	166	1.95		2.35	2.42	2.27	2.47		2.68	2.76
LSTM-3LR(ICCV'15)	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(CVPR'16)	0.97	1.14		1.35	1.46	0.97	1.14		1.35	2.08	1.22	1.49		1.83	1.93
RRNN(CVPR'17)	0.29	0.52	0.67	0.87	1.10	0.36	0.67	0.97	1.18	1.27	0.41	0.90	1.14	1.31	1.40
CSS(CVPR'18)	0.23	0.37	0.47	0.57	0.71	0.26	0.48	0.74	0.95	0.96	0.31	0.66	0.88	0.96	1.05
SkelNet(Ours)	0.23	0.39	0.50	0.58	0.71	0.26	0.47	**0.70**	**0.91**	**0.89**	**0.29**	**0.62**	0.88	**0.89**	1.00
Skel-TNet(Ours)	**0.21**	**0.34**	**0.43**	**0.55**	**0.70**	**0.25**	**0.46**	0.71	**0.91**	0.90	0.30	0.64	**0.86**	0.90	**0.99**

Table 2: Detailed results reported in Euler angle error on representative activities of the Human3.6M dataset, for short-term (80, 160, 240, 320 and 400ms) prediction. Bold means the best performance.

	Human3.6M	CMU mocap dataset
full	**1.17**	0.89
w long prior	1.19	**0.87**
w tanh	1.19	0.92
w/o residual connection	2.48	1.77
w/o brances	1.23	0.94
w/o branches w/o LReLUs	1.47	1.08
w/o branches w/o LReLUs w/o dropout	1.53	1.16
SkelNet_UD	1.19	0.93
SKelNet_LR	1.21	0.94

Table 3: Ablation studies on different components in the network design of our SkelNet, results reported is the mean value of MoF for all activities.

Noise Variance	Human3.6M 0.1	0.3	0.5	CMU mocap dataset 0.1	0.3	0.5	Total parameters
RRNN	1.65	1.74	1.87				$\approx 0.5m$
CSS	1.26	**1.50**	1.82	1.03	1.21	1.50	$\approx 9m$
SkelNet	**1.24**	**1.50**	**1.80**	**0.93**	**1.15**	**1.41**	$\approx 0.3m$

Table 4: The averaging prediction error reported in MoF on two datasets, the input is under Gaussian noise with different variance [0.1, 0.3, 0.5]. The rightmost column records total number of parameters used in each method.

	Human3.6M Walk	Eat	Discuss	Avg	CMU mocap dataset Run	BSignal	Soccer	Avg
SkelNet	**0.49**	0.46	**0.70**	**0.76**	**0.49**	0.38	**0.48**	0.60
C-RNN	0.55	**0.41**	**0.70**	**0.76**	0.59	**0.27**	0.52	**0.59**
C-RNN(w/o)	0.56	0.45	0.85	0.79	0.59	0.33	0.64	0.63

Table 5: Ablation studies on Skel-TNet's components. All MoF reported are for specific activities and averaging value computed on all activities of two datasets. C-RNN and C-RNN(w/o) denote C-RNN trained by $\mathcal{L}_{conv}$ and *Sampling-based loss*.

	Human3.6M	CMU mocap dataset
Full	**0.73**	**0.55**
w/o T	0.78	0.60
w/o S	0.79	0.59
w/o M	0.80	0.62
Trained jointly(1)	0.82	0.69
Trained jointly(2)	0.83	0.67

Table 6: Ablation studies on Skel-TNet's architecture. Results reported is the mean value of MoF for all activities. Trained jointly (1) and (2) represent arranging Skel-TNet alternatives in two different orders.

and testing is under Gaussian noise with different variances ([0.1, 0.3, 0.5]). Table 4 shows that SkelNet has advantage in terms of prediction accuracy to two competitors in the CMU mocap dataset, no matter what variance of Gaussian noise has been used. However, it is only able to perform comparably with CSS on the Human3.6M dataset. Margins in both datasets between SkelNet and CSS are not increased as variance goes up. However, safe conclusion can be drawn that SkelNet is slightly more robust than its competitors.

Skel-TNet component: We exam the performance of Skel-TNet's components, training them independently for short-term prediction and results are reported in Table 5. Generally, SkelNet and C-RNN are able to produce predictions with roughly same MoF in averaging results computed over all activities. However, SkelNet has advantages to C-RNN on activities like "WashWindow" and "Walking", conversely, C-RNN achieve the more accurate prediction on "BasketballSignal" and "Eating". Moreover, our $\mathcal{L}_{conv}$ do provide C-RNN with improvement in its generalization capability, as shown in prediction results on averaging MoF over all activities and MoF on some certain activities ("Soccer" and "Discussion").

Skel-TNet architecture: The Skel-TNet is a multi-staged

processing network, there are several alternatives to its configuration. We firstly test Skel-TNet's generalization ability when Merging Network is merely with one component, SkelNet (w/o T) or C-RNN (w/o S), by removing the other. Furthermore, we remain both of SkelNet and C-RNN but replace Merging Network with averaging sum for two input sequences (w/o M). At the end, we train three components in stacked way, orders are SkelNet, C-RNN and Merging Network, or C-RNN, SkelNet and Merging Network. We get rid of one of two trainable weights in Merging Network applied on input sequences, when there is only one sequence being input to Merging Network. All results are in Table 6.

Conclusion

We propose SkelNet for human motion prediction at first. Unlike the previous work, we pay attention to different dynamic patterns from local components of the human pose, this is achieved by dividing the human pose into five parts that then be fed into five component-specific layers of Skel-Net for obtaining representations of local structures. Our SkelNet is designed as effective yet simple (1/30 parameter number of the state-of-the-art method) and can be strengthened for its capability on learning temporal dynamics in the

Skel-TNet, which is our the second proposed network for prediction task. Both SkelNet and Skel-TNet obtain superior or at least comparable performances to recently proposed methods, as shown by experimental results.

References

Arikan, O.; Forsyth, D. A.; and O'Brien, J. F. 2003. Motion synthesis from annotations. In *ACM Transactions on Graphics (TOG)*, volume 22, 402–408. ACM.

Butepage, J.; Black, M. J.; Kragic, D.; and Kjellstrom, H. 2017. Deep representation learning for human motion prediction and classification. In *CVPR*.

Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Fang, H.; Xu, Y.; Wang, W.; Liu, X.; and Zhu, S.-C. 2018. Learning knowledge-guided pose grammar machine for 3d human pose estimation. In *AAAI*.

Fragkiadaki, K.; Levine, S.; Felsen, P.; and Malik, J. 2015. Recurrent network models for human dynamics. In *ICCV*.

Ghosh, P.; Song, J.; Aksan, E.; and Hilliges, O. 2017. Learning human motion models for long-term predictions. In *3DV*.

Girshick, R.; Donahue, J.; Darrell, T.; and Malik, J. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Huang, X.; Li, Y.; Poursaeed, O.; Hopcroft, J.; and Belongie, S. 2017. Stacked generative adversarial networks. In *CVPR*.

Ionescu, C.; Papava, D.; Olaru, V.; and Sminchisescu, C. 2014. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *TPAMI* 36(7):1325–1339.

Jain, A.; Zamir, A. R.; Savarese, S.; and Saxena, A. 2016. Structural-rnn: Deep learning on spatio-temporal graphs. In *CVPR*.

Koppula, H. S., and Saxena, A. 2016. Anticipating human activities using object affordances for reactive robotic response. *TPAMI* 38(1):14–29.

Lab, C. G. *http://mocap.cs.cmu.edu/, CMU motion capture data*.

Lamb, A. M.; GOYAL, A. G. A. P.; Zhang, Y.; Zhang, S.; Courville, A. C.; and Bengio, Y. 2016. Professor forcing: A new algorithm for training recurrent networks. In *NIPS*.

Li, C.; Zhang, Z.; Sun Lee, W.; and Hee Lee, G. 2018. Convolutional sequence to sequence model for human dynamics. In *CVPR*.

Lin, X., and Amer, M. R. 2018. Human motion modeling using dvgans. *arXiv:1804.10652*.

Liu, J.; Shahroudy, A.; Xu, D.; and Wang, G. 2017a. Spatio-temporal lstm with trust gates for 3d human action recognition. In *ECCV*.

Liu, J.; Wang, G.; Hu, P.; Duan, L.-Y.; and Kot, A. C. 2017b. Global context-aware attention lstm networks for 3d action recognition. In *CVPR*.

Liu, C. K.; Hertzmann, A.; and Popović, Z. 2005. Learning physics-based motion style with nonlinear inverse optimization. In *ACM Transactions on Graphics (TOG)*, volume 24, 1071–1081. ACM.

L'opez-M'endez, A.; Gall, J.; Casas, J.; and van Gool, L. 2012. Metric learning from poses for temporal clustering of human motion. In *British Machine Vision Conference (BMVC)*.

Maas, A. L. 2013. Rectifier nonlinearities improve neural network acoustic models.

Martinez, J.; Black, M. J.; and Romero, J. 2017. On human motion prediction using recurrent neural networks. In *CVPR*.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: A simple way to prevent neural networks from overfitting. *JMLR* 15:1929–1958.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

Taylor, G. W.; Sigal, L.; Fleet, D. J.; and Hinton, G. E. 2010. Dynamical binary latent variable models for 3d human pose tracking. In *CVPR*.

Taylor, G. W.; Hinton, G. E.; and Roweis, S. T. 2007. Modeling human motion using binary latent variables. In *NIPS*.

Tekin, B.; Rozantsev, A.; Lepetit, V.; and Fua, P. 2016. Direct prediction of 3d body poses from motion compensated sequences. In *CVPR*.

Wang, J. M.; Fleet, D. J.; and Hertzmann, A. 2005. Gaussian process dynamical models. In *NIPS*.

Williams, R. J., and Zipser, D. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation* 1(2):270–280.

Wu, D., and Shao, L. 2014. Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition. In *CVPR*.

Xu, C.; He, J.; Zhang, X.; Yao, C.; and Tseng, P.-H. 2018. Geometrical kinematic modeling on human motion using method of multi-sensor fusion. *Information Fusion* 41:243–254.

Yu, X., and Porikli, F. 2017. Hallucinating very low-resolution unaligned and noisy face images by transformative discriminative autoencoders. In *CVPR*.

Zhang, H.; Xu, T.; Li, H.; Zhang, S.; Wang, X.; Huang, X.; and Metaxas, D. 2017. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *arXiv:1710.10916*.

Zhao, R., and Ji, Q. 2018. An adversarial hierarchical hidden markov model for human pose modeling and generation. In *AAAI*.