

# Action-Agnostic Human Pose Forecasting

Hsu-kuang Chiu, Ehsan Adeli, Borui Wang, De-An Huang, Juan Carlos Niebles  
Stanford University

{hkchiu, eadeli, wbr, dahuang, jniebles}@cs.stanford.edu

## Abstract

Predicting and forecasting human dynamics is a very interesting but challenging task with several prospective applications in robotics, health-care, etc. Recently, several methods have been developed for human pose forecasting; however, they often introduce a number of limitations in their settings. For instance, previous work either focused only on short-term or long-term predictions, while sacrificing one or the other. Furthermore, they included the activity labels as part of the training process, and require them at testing time. These limitations confine the usage of pose forecasting models for real-world applications, as often there are no activity-related annotations for testing scenarios. In this paper, we propose a new action-agnostic method for short- and long-term human pose forecasting. To this end, we propose a new recurrent neural network for modeling the hierarchical and multi-scale characteristics of the human dynamics, denoted by triangular-prism RNN (TP-RNN). Our model captures the latent hierarchical structure embedded in temporal human pose sequences by encoding the temporal dependencies with different time-scales. For evaluation, we run an extensive set of experiments on Human 3.6M and Penn Action datasets and show that our method outperforms baseline and state-of-the-art methods quantitatively and qualitatively. Codes are available at [https://github.com/eddyhkchiu/pose\\_forecast\\_wacv/](https://github.com/eddyhkchiu/pose_forecast_wacv/)

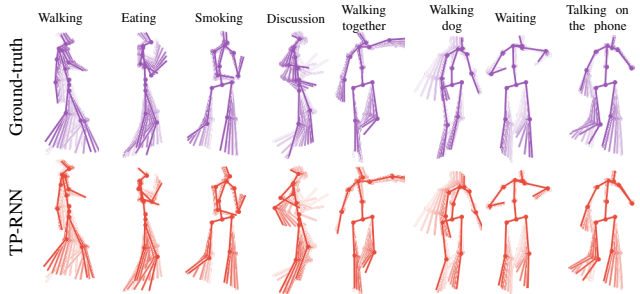


Figure 1: Ground-truth pose sequences (first row) and forecasted ones by our method (second row). Solid colors indicate later time-steps and faded ones are older. The course of changes in the predicted and ground-truth poses resemble similar patterns. Furthermore, body-part movement patterns show that different parts depend on each other, but with varied temporal scales. Hence, hierarchical multi-scale modeling may encode the latent structures of human dynamics.

as poses [33]. Predicting human dynamics is hence delineated by predicting the course of changes in human poses [11, 18, 24, 36, 40, 43].

Detecting and predicting poses has long been an interesting topic in the computer vision community [27, 33, 43]. Recently, several methods have been introduced for forecasting human poses in the near future [11, 18, 24, 36, 40]. In a recent work, Martinez *et al.* [23] noted that, although great advancements in pose forecasting has been achieved by prior works, they often fail to generate realistic human poses, especially in short-term predictions, and in most cases they fail to even outperform the zero-velocity predictor (*i.e.*, repeating the very last seen pose as predictions for the future). Ghosh *et al.* [16], with reference to [23], attributed this finding to the side-effects of curriculum learning (such as in [11]), commonly practiced for temporal forecasting. With such observations, some previous works focused on short-term forecasting of human poses [18, 23], and some others exclusively aimed attention at long-term predictions [8, 16, 36]. However, most of the previous methods achieve reasonable performance by incorporating action labels as extra data annotation in their models, *i.e.*, they either trained pose fore-

## 1. Introduction

Humans are able to predict how their surrounding environment may change and how other people move. This inclination and aptitude is crucial to make social life and interaction with others attainable [4]. As such, to create machines that can interact with humans seamlessly, it is very important to convey the ability of predicting short- and long-term future of human dynamics based on the immediate present and past. Recently, computer vision researchers attempted predicting human dynamics from images [11], or through time in videos [8, 31, 37]. Human dynamics are mainly defined as a set of structured body joints, known

casters on each action class separately (e.g., [15, 18]) or incorporated the action labels as an extra input to the model and concluded that including action labels improves the results (e.g., [23]). Although action labels are not hard to acquire for training samples, but the use of labels for testing videos as an input of the model is *unrealistic* and makes the introduced models *unusable* for real-world applications, as action labels are not available during testing [1]. Unlike these previous works, our method learns a pose forecaster regardless of their action class. We propose an action-agnostic model for pose forecasting by implicitly encoding the short- and long-term dependencies within actions.

In this paper, we propose a new recurrent neural network (RNN) model for forecasting human poses in both short- and long-term settings. To model human dynamics and to capture the latent hierarchical structure in the temporal pose sequences, we **encode the temporal dependencies of different time-scales in a hierarchical interconnected sequence of RNN cells**. Called *Triangular-Prism Recurrent Neural Network (TP-RNN)*, our proposed method contains a new multi-phase hierarchical multi-scale RNN architecture that is tailored for modeling human dynamics in visual scenes. Different from the original hierarchical multi-scale RNNs (HM-RNN) for representation of natural language sequences [12], our architecture redefines hierarchies and multi-scale interconnections to accommodate human dynamics. Sequences of human poses through time involve hierarchical and multi-scale structures, as movements of different body-parts (and joints) depend on each other. Besides, each of these parts (and joints) have distinct motion patterns and hence different temporal scales for particular activities. For instance, during ‘walking’, arms and legs move in a shorter temporal scale (i.e., more frequently) compared to the torso, which is potentially in a longer temporal scale (see Fig. 1). Learning the hierarchical multi-scale dynamics of changes in human poses enables TP-RNN to construct an implicit encoding of short- and long-term dependencies within action classes, and hence be able to predict future sequences without the demand for the supervising signal from action labels.

Our model **takes pose velocities** (i.e., differences between the current and the immediate previous poses, ignoring the division by the constant time duration between two consecutive frames) as inputs and **outputs predictions in the same space of velocities**. As opposed to the previous works [15, 18, 40] that focused on predicting sequences of poses (as structured objects) in the forms of either joint angles (e.g., [15]) or joint locations (such as in [18]), we argue that **forecasting in the velocities space boosts prediction power since human poses change slightly in narrow time-steps**. Different from [23], in which residual connections were applied on top of RNN (using residuals as outputs only, while inputs are poses), our method uses velocities as both inputs and outputs and shows significantly improved forecasting results.

To evaluate the proposed method, we run an extensive set of experiments on Human 3.6M [17] and Penn Action [46] datasets, and compare the results with several baseline and state-of-the-art algorithms on these datasets. The comparison shows that our method outperforms others in terms of the **mean angle error (MAE)** on Human 3.6M and the Percentage of Correct Keypoint (PCK) score on Penn Action. Our action-agnostic method leads to superior results in cases of both short- and long-term predictions (some are visualized in Fig. 1) even in comparison to the methods designed specifically for short- or long-term predictions or methods that use action labels as inputs to their models.

In summary, the contributions of this paper are three-fold: (1) we propose an action-agnostic model that trains the pose forecaster regardless of action classes; (2) we propose a new model, TP-RNN, inspired by the hierarchical multi-scale RNN from NLP research, for forecasting human dynamics. TP-RNN implicitly encodes the action classes and, unlike previous methods, does not require external action labels during training; (3) we show that operating in the velocity space (i.e., using pose velocities as both inputs and outputs of the network) improves the results of our model.

## 2. Related Works

In this Section, we review the relevant literature on human motion, activity, and pose forecasting, along with the previous works on hierarchical and multi-scale RNNs (and Long Short-Term Memory cells, i.e., LSTMs).

**Predicting Motion and Human Dynamics:** The majority of the recent works on motion representation has mainly focused on anticipating the future at the pixel level. For instance, generative adversarial networks (GANs) were used to generate video pixels [25, 37], and RNNs for anticipating future video frames [22]. To predict dense trajectories, Walker *et al.* [39] used a CNN, and others have used random forests [28] or variational auto-encoders [38]. Other works targeted predicting the future in forms of semantic labels (e.g., [20, 21, 38]) or activity labels (e.g., [2, 6, 34, 40]). Human dynamics, however, could be better characterized by 2D [10, 29, 30] or 3D [3, 26, 35, 45] poses, and several works attempted to detect these poses from images or videos [32, 35, 45]. Modeling human motions is commonly defined in two different ways: *probabilistic and state transition models* (such as Bayesian and Gaussian processes [42] or hidden Markov models [44]), and *deep learning methods*, in particular RNNs and LSTMs, e.g., [16, 18, 23]. For instance, Jain *et al.* [18] proposed a structural RNN to cast an arbitrary spatio-temporal graph as a RNN and use it for modeling human pose in temporal video sequences. In this work, we propose a new multi-phase hierarchical multi-scale RNN for modeling human dynamics to forecast poses.

**Human Pose Forecasting:** Forecasting human poses in images and video sequences is relatively new compared to

predicting image or video pixels. Although it can be a very useful task with great applications (*e.g.*, in predictive surveillance, patient monitoring, *etc.*), just recently researchers have paid more attention to it [8, 11, 18, 23, 24, 36, 40].

Specifically, Chao *et al.* [11] proposed a 3D Pose Forecasting Network (3D-PFNet) for forecasting human dynamics from static images. Their method integrates recent advances on single-image human pose estimation and sequence prediction. In another work, [24] introduced a method to predict 3D positions of the poses, given their 2D locations. Barsoum *et al.* [8] proposed a sequence-to-sequence model for the task of probabilistic pose prediction, trained with an improved Wasserstein GAN [5]. Walker *et al.* [40] proposed a method based on variational autoencoders and GANs to predict possible future human movements (*i.e.*, poses) and then predict future frames. Fragkiadaki *et al.* [15] proposed two architectures for the task of pose prediction, one denoted by LSTM-3LR (3 layers of LSTM cells) and the second one as ERD (Encoder-Recurrent-Decoder). These two models are based on a sequence of LSTM units. Martinez *et al.* [23] used a variation of RNNs to model human motion with the goal of learning time-dependent representations for human motion prediction synthesis in a short-term. Three key modifications to recent RNN models were introduced, in the architecture, loss function, and the training procedures. In another work, Bütetpage *et al.* [9] proposed an encoding-decoding network that learns to predict future 3D poses from the immediate past poses, and classify the pose sequences into action classes. These two methods incorporate a high-level supervision in the form of action labels, which itself improves the performance. However, in many real world applications of human motion analysis there are no motion or activity labels available during inference time.

**Hierarchical Multi-Scale RNNs:** Our proposed architecture is inspired by the hierarchical multi-scale recurrent neural networks (HM-RNN) introduced in [12]. HM-RNN builds on multi-scale RNNs [19] that model high-level abstraction changes slowly with temporal coherency while low-level abstraction has quickly changing features sensitive to the precise local timing [14]. This architecture is able to learn the latent representation of natural language sequences in different hierarchies (*e.g.*, words, phrases, and sentences) to build character-level language models for predicting future sequences [7, 14]. We observe that multi-scale temporal information at different hierarchical levels can be beneficial in modeling human dynamics. However, it is difficult to adopt this approach directly because we do not have clear-cut temporal boundaries as in natural language data.

### 3. Triangular-Prism RNN (TP-RNN)

As discussed earlier, sequences of human poses can be subsumed under hierarchical and multi-scale structures, since movements of different body parts hinge on movements

of other parts. Also, each part often has distinct motion patterns and hence different temporal scales when performing particular activities. Therefore, in contrast to the classical single-layer LSTM or RNN architectures (such as in [18]) or stacked LSTMs (*e.g.*, in [13, 41]), we introduce **multi-phase hierarchical multi-scale upper layers** of LSTM sequences to better learn the longer-term temporal relationships between different time-steps in a series of different granularities.

The inputs and outputs of the model, as mentioned earlier, are velocities. Let the pose in time  $t$  be identified by  $P_t$ , then the velocity in time  $t$  can be defined as  $V_t = P_t - P_{t-1}$ . Therefore, for any given granularity coefficient  $K$  and the number of levels  $M$ , we define a multi-phase hierarchical multi-scale RNN with scale  $K$  and  $M$  levels. On the first level, we have a regular LSTM sequence taking the velocity information at each time-step as inputs. Then, on the second level, we define  $K$  different sequences of LSTM units, with each sequence only taking the inputs from the LSTM units on the first level at time-steps that are congruent modulo  $K$ . For example, if  $K = 2$ , then we have two LSTM sequences at level 2, with the first one taking inputs from  $t = \{1, 3, 5, \dots\}$  and the second one from  $t = \{2, 4, 6, \dots\}$  (see Fig. 2 for illustrations). Note that these LSTMs in the same level of the hierarchy share weights and this shifting scheme is actually used as a data augmentation strategy to learn longer-term dependencies in a more reliable way. Similarly, if we define a third level in the hierarchy, for each of the  $K$  LSTM sequences on the second level, we will have  $K$  different LSTM sequences each taking congruent-modulo- $K$  inputs from it, resulting in a total of  $K^2$  LSTM sequences in the third level. This process of spawning new higher-level LSTM sequences over the hierarchy continues for  $M - 1$  levels, which will have  $K^{M-1}$  LSTM sequences in the  $M$ th level. Therefore, logically, we have a total of  $(K^{M-1} + K^{M-2} + \dots + K^3 + K^2 + K + 1)$  LSTM sequences in the whole architecture, while only  $M$  different ones are kept physically (since the LSTM sequences in each level share weights). For the sampling stage, we introduce a two-layer fully-connected network to generate the velocity predictions given the velocity at the current time-step and the corresponding hidden units across all hierarchies. Fig. 2 illustrates an example of our architecture for  $K = M = 2$ .

Our hierarchical model is inspired by HM-RNN [12], however, as discussed earlier, we cannot directly apply HM-RNN to our task, due to the differences between human dynamics and natural language sequences. TP-RNN models short-term dependencies in lower levels of the hierarchy and long-term temporal dependencies in higher levels, and hence can capture the latent hierarchical structure in different time-scales. Different from HM-RNN, since human dynamics (unlike language models) do not have natural boundaries in the sequences, TP-RNN uses the outputs of all hierarchy levels to represent the sequence and predict the next

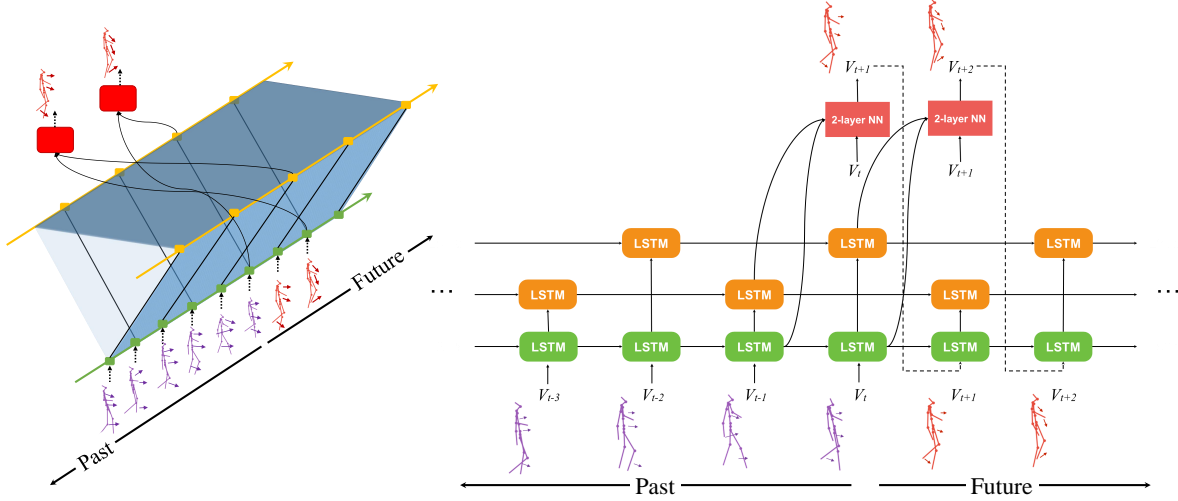


Figure 2: Architecture of TP-RNN with  $K = 2$  and  $M = 2$ . Left: 3D view of the triangular-prism RNN. Right: 2D projection.

element in the sequence. Moreover, instead of having only one RNN layer in each level of the hierarchies, as we move up in the hierarchy, TP-RNN decreases the resolution by one unit but has multiple RNN layers in the higher levels, to capture the temporal dynamics from different phases of its lower hierarchy level. All RNN layers in each single hierarchy level share parameters; although this scheme does not increase model parameters, shifting phases from each lower level to create their immediate higher level RNNs helps in augmenting the data and learning better models at each level. Therefore, as we go up in the hierarchy, more parallel RNN layers are incorporated, and hence we chose the name triangular-prism RNN (see Fig. 2). This architecture design provides the following advantages over HM-RNN for modeling human dynamics: (1) the lowest layer RNN can learn the finest grained scale motion dynamics without the interference from the higher levels, and higher levels capture different characteristics of the dynamics each at a certain scale; (2) during the prediction of each time-step, we have the most up-to-date RNN outputs from different hierarchies, each of which carries temporal motion information with different scales. On the contrary, HM-RNN does not provide the most up-to-date larger scale temporal information when the current prediction time-step is not right after a boundary.

## 4. Experiments

We evaluate our method on two challenging datasets. The results are analyzed and compared with baseline and state-of-the-art techniques, both quantitatively and qualitatively.

In our architecture, we use LSTMs with hidden size 1024 as the RNN cells (the orange and the green blocks in Fig. 2). For the final pose velocity generation networks (the red blocks in Fig. 2), we use 2 fully-connected layers with hidden sizes 256 and 128, followed by a Leaky-ReLU non-

linearity layer. The training setup is similar to [23]. The optimization uses mini-batch stochastic gradient descent with batch size 16, clipping the gradients up to  $\ell_2$ -norm value of 5. The learning rate is initialized to 0.01 and decayed along the training iterations. We train for 100,000 iterations and record the performance coverage in the end.

### 4.1. Datasets

To test the performance of our model for human pose forecasting, we run extensive experiments using Human 3.6M [17] and Penn Action [46] datasets.

**Human 3.6M dataset:** The Human 3.6M dataset [17] is one of the largest publicly available datasets of human motion capture data. This dataset contains video sequences of a total of 15 different human activity categories, each performed by seven actors in two different trials. The videos were recorded at 50Hz (*i.e.*, 20ms between each two consecutive pose frames). Following previous work [18, 23], in our experiments, we downsample the pose sequence by 2. In the dataset, each pose is represented as exponential map representations of 32 human joints in the 3D space, and during evaluation, we employ the measurement of the Euclidean distance between the ground-truth pose and our predicted pose in the angle space as the error metric. Consistent with the previous work, we also use Subject 5 as the test data and Subjects 1, 6, 7, 8, 9, 11 as training. Similar to [18, 23], we train our models using the past 50 frames (2000ms) as the input sequence, and forecast the future 25 frames (1000ms). The training loss is calculated by the mean angle error (MAE) from each of the predicted future frames.

**Penn Action dataset:** The second dataset we experiment on is the Penn Action dataset [46], which contains 2326 video sequences of 15 different actions and human joint annotations for each sequence. Each human pose is represented by



13 human joint coordinates in the 2D space. Following the same data split of [11, 46], 1258 video sequences are used for training and the remaining 1068 video sequences are used for testing. For this dataset, the previous state-of-the-art, 3D-PFNet [11], takes the first frame image as input, and outputs the poses extracted from that frame and the future 15 frames, resulting in total of 16 frame poses. The model performance is evaluated using PCK@0.05 [11]. For the experiments on this dataset, we use a single pose in a past frame (ignoring the frame image) as the input, and the outputs are the predictions of poses of the future 16 frames. Although the input format of [11] is slightly different from ours, it is still a fair comparison of pose forecasting capabilities.

## 4.2. Results

**Baseline Methods:** We use the following recent research to compare with: ERD [15], LSTM-3LR [15], SRNN [18], Dropout-AutoEncoder [16], 3D-PFNet [11], and Residual [23]. Similar to [23], we include the zero-velocity model as a naïve baseline for comparison. We also include our implementations of different LSTM-based models as part of the comparison (*i.e.*, conducting ablation tests).

First set of our experiments compares the single layer LSTM model with pose as the input (denoted by Single Layer (Pose)) and the same model but with velocity as the input (Single Layer (Vel.)), to demonstrate that conducting the experiments in the velocity space and feeding it into LSTM sequences can better capture the human motion dynamics. As mentioned earlier, when both inputs and outputs are all velocities with similar small numerical scales, it is easier for the model to be trained. In the second set of experiments, we build multiple 2-Layer LSTM models with different architectures using velocity as the input, including the most basic one that simply stacks 2 layers of LSTMs (Stacked 2-Layer (Vel.)), commonly called multi-layer LSTM [13, 41]. On top of the basic model, we build further extensions with hierarchical and multi-scale structures: two independent LSTMs (Double-scale (Vel.)), which, unlike the regular multi-layer LSTM, its higher level one does not use the output from the lower level as the input. Instead, the higher level LSTM’s input is the larger scale of velocity, *i.e.*, the velocity calculated by the pose sequence only at the odd time-steps, or only at the even time-steps. The next model (Double-scale (Hier., Vel.)) is similar to HM-RNN [12], but with slight modification of setting the higher level LSTM scale to a constant number 2, due to the fact that there is no natural boundary in human motion sequences. Another model (Double-scale (Phase, Vel.)) has multiple phases in the higher level LSTMs, capturing larger scale velocity information, rather than using the lower level LSTM outputs. Finally, we implement our proposed model (TP-RNN) with double scale setting. Note, to showcase the superiority of the proposed technique we report the results for  $K = M = 2$  in TP-RNN, which

demonstrates that without the need to increase the network parameters, our network already outperforms all other methods. However, we also conduct an experiment for analyzing the effect of the number of levels in TP-RNN, and show models with more hierarchies can lead to even better results.

**Comparison on Human 3.6M dataset:** Previous literature published their performance numbers on either short-term (up to 400ms) [23] or long-term (up to 1000ms) [18] predictions. Besides, some of them only report the prediction on a small set of actions (*i.e.*, ‘walking’, ‘eating’, ‘smoking’, and ‘discussion’) [15, 18], while others report the results for all 15 actions [23] in the Human 3.6M dataset [17]. To compare with all the above different settings, for each of our architectures, we train a single action-agnostic model using sequence data from all of the 15 actions, without any supervision from the ground-truth action labels. We use the loss over each forecasted frame up to 1000ms (25 frames). We follow the settings of [18, 23] for the length of the input seed observed pose (*i.e.*, 2000ms, 50 frames).

Table 1 shows the MAE for ‘walking’, ‘eating’, ‘smoking’, and ‘discussion’ for short-term predictions. Our model (TP-RNN) outperforms all the baseline results, including the current state-of-the-art, Residual model [23], in short-term forecasting. In the Residual model [23], pose information is used to predict the velocity of the next frame. Note that the numerical scale of velocity is much smaller compared to the pose. On the contrary, in our proposed model, velocity information of the past is fed into the models to predict the next velocity. Therefore, the scales of inputs and outputs are the same, which potentially puts the neural network in an easier path to train. For actions with large movements (like ‘Walking’ and ‘Eating’) our model outperforms the baselines and the state-of-the-art by a large margin. However, like other previous methods, our method has hard time to forecast ‘difficult-to-predict’ actions (like in ‘Smoking’ and ‘Discussion’). Although, our results in those activities are also superior to all other methods, they are close to the zero-velocity baseline. Our proposed TP-RNN is setting a new state-of-the-art for pose forecasting on this dataset without the need of action labels at test time. Furthermore, it conducts both short- and long-term predictions simultaneously without sacrificing the accuracy of either end.

Table 2 shows the MAE for the same set of the four actions in the long-term prediction task. The state-of-the-art model (*i.e.*, Residual [23]) does not report the long-term prediction performance results, therefore we use its open-source implementation code to collect the results for long-term predictions. Note that when changing the training loss from short-term predictions to long-term predictions, this model sacrifices the prediction accuracy in the short-term time-range (less than 400ms) in order to gain the extra long-term (400ms to 1000ms) prediction ability.

The long-term prediction of the Residual model [23] still

Table 1: MAE for four action classes in the short-term forecasting experiment (prior-work results from [23]). In each column, the best obtained results are typeset in boldface and the second best are underlined. AA: Action-Agnostic, N/A: Not Applicable.

|               | AA  | Walking     |             |             |             | Eating      |             |             |             | Smoking     |             |             |             | Discussion  |             |             |             |
|---------------|-----|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| milliseconds  |     | 80          | 160         | 320         | 400         | 80          | 160         | 320         | 400         | 80          | 160         | 320         | 400         | 80          | 160         | 320         | 400         |
| ERD [15]      | ✗   | 0.93        | 1.18        | 1.59        | 1.78        | 1.27        | 1.45        | 1.66        | 1.80        | 1.66        | 1.95        | 2.35        | 2.42        | 2.27        | 2.47        | 2.68        | 2.76        |
| LSTM-3LR [15] | ✗   | 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 [18]     | ✗   | 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        |
| Residual [23] | ✗   | <u>0.28</u> | <u>0.49</u> | <u>0.72</u> | <u>0.81</u> | <u>0.23</u> | <u>0.39</u> | <u>0.62</u> | <u>0.76</u> | <u>0.33</u> | 0.61        | 1.05        | 1.15        | <u>0.31</u> | 0.68        | 1.01        | <u>1.09</u> |
| Zero-velocity | N/A | 0.39        | 0.68        | 0.99        | 1.15        | 0.27        | 0.48        | 0.73        | 0.86        | <b>0.26</b> | <u>0.48</u> | <u>0.97</u> | <u>0.95</u> | <u>0.31</u> | <u>0.67</u> | <b>0.94</b> | <b>1.04</b> |
| TP-RNN (Ours) | ✓   | <b>0.25</b> | <b>0.41</b> | <b>0.58</b> | <b>0.65</b> | <b>0.20</b> | <b>0.33</b> | <b>0.53</b> | <b>0.67</b> | <b>0.26</b> | <b>0.47</b> | <b>0.88</b> | <b>0.90</b> | <b>0.30</b> | <b>0.66</b> | <u>0.96</u> | <b>1.04</b> |

Table 2: MAE for four action classes in the long-term forecasting experiments (prior-work results from SRNN [18], Dropout-AutoEncoder [16], and code from [23]). In each column, the best obtained results are typeset with boldface and the second best are underlined. AA: Action-Agnostic, N/A: Not Applicable, Dropout-AE: Dropout-AutoEncoder.

|                 | AA  | Walking     |             |             |             |             | Eating      |             |             |             |             | Smoking     |             |             |             |             | Discussion  |             |             |             |             |
|-----------------|-----|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| milliseconds    |     | 80          | 160         | 320         | 560         | 1000        | 80          | 160         | 320         | 560         | 1000        | 80          | 160         | 320         | 560         | 1000        | 80          | 160         | 320         | 560         | 1000        |
| ERD [15]        | ✗   | 1.30        | 1.56        | 1.84        | 2.00        | 2.38        | 1.66        | 1.93        | 2.88        | 2.36        | 2.41        | 2.34        | 2.74        | 3.73        | 3.68        | 3.82        | 2.67        | 2.97        | 3.23        | 3.47        | 2.92        |
| LSTM-3LR [15]   | ✗   | 1.18        | 1.50        | 1.67        | 1.81        | 2.20        | 1.36        | 1.79        | 2.29        | 2.49        | 2.82        | 2.05        | 2.34        | 3.10        | 3.24        | 3.42        | 2.25        | 2.33        | 2.45        | 2.48        | 2.93        |
| SRNN [18]       | ✗   | 1.08        | 1.34        | 1.60        | 1.90        | 2.13        | 1.35        | 1.71        | 2.12        | 2.28        | 2.58        | 1.90        | 2.30        | 2.90        | 3.21        | 3.23        | 1.67        | 2.03        | 2.20        | 2.39        | 2.43        |
| Dropout-AE [16] | ✗   | 1.00        | 1.11        | 1.39        | 1.55        | 1.39        | 1.31        | 1.49        | 1.86        | 1.76        | 2.01        | 0.92        | 1.03        | 1.15        | 1.38        | 1.77        | 1.11        | 1.20        | 1.38        | 1.53        | <b>1.73</b> |
| Residual [23]   | ✗   | <u>0.32</u> | <u>0.54</u> | <u>0.72</u> | <u>0.86</u> | <u>0.96</u> | <u>0.25</u> | <u>0.42</u> | <u>0.64</u> | <u>0.94</u> | <u>1.30</u> | <u>0.33</u> | <u>0.60</u> | 1.01        | 1.23        | 1.83        | 0.34        | 0.74        | 1.04        | 1.43        | 1.75        |
| Zero-velocity   | N/A | 0.39        | 0.68        | 0.99        | 1.35        | 1.32        | 0.27        | 0.48        | 0.73        | 1.04        | 1.38        | <b>0.26</b> | <b>0.48</b> | <u>0.97</u> | <u>1.02</u> | <u>1.69</u> | <u>0.31</u> | <u>0.67</u> | <b>0.94</b> | <u>1.41</u> | 1.96        |
| TP-RNN (Ours)   | ✓   | <b>0.25</b> | <b>0.41</b> | <b>0.58</b> | <b>0.74</b> | <b>0.77</b> | <b>0.20</b> | <b>0.33</b> | <b>0.53</b> | <b>0.84</b> | <b>1.14</b> | <b>0.26</b> | <b>0.48</b> | <b>0.88</b> | <b>0.98</b> | <b>1.66</b> | <b>0.30</b> | <b>0.66</b> | <u>0.98</u> | <b>1.39</b> | 1.74        |

Table 3: Long-term forecasting MAE comparison for the remaining 11 actions in Human 3.6 dataset.

|          | Directions  |             |             |             |             |             | Greeting    |             |             |             |             |             | Talking on the phone |             |             |             |             |             | Posing            |             |             |             |             |             |
|----------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|----------------------|-------------|-------------|-------------|-------------|-------------|-------------------|-------------|-------------|-------------|-------------|-------------|
| millisec | 80          | 160         | 320         | 400         | 560         | 1000        | 80          | 160         | 320         | 400         | 560         | 1000        | 80                   | 160         | 320         | 400         | 560         | 1000        | 80                | 160         | 320         | 400         | 560         | 1000        |
| [23]     | 0.44        | 0.69        | 0.83        | 0.94        | 1.03        | 1.49        | 0.53        | 0.88        | 1.29        | 1.45        | <b>1.72</b> | 1.89        | 0.61                 | 1.12        | 1.57        | 1.74        | 1.59        | 1.92        | 0.47              | 0.87        | 1.49        | 1.76        | 1.96        | <b>2.35</b> |
| TP-RNN   | <b>0.38</b> | <b>0.59</b> | <b>0.75</b> | <b>0.83</b> | <b>0.95</b> | <b>1.38</b> | <b>0.51</b> | <b>0.86</b> | <b>1.27</b> | <b>1.44</b> | <b>1.72</b> | <b>1.81</b> | <b>0.57</b>          | <b>1.08</b> | <b>1.44</b> | <b>1.59</b> | <b>1.47</b> | <b>1.68</b> | <b>0.42</b>       | <b>0.76</b> | <b>1.29</b> | <b>1.54</b> | <b>1.75</b> | 2.47        |
|          | Purchases   |             |             |             |             |             | Sitting     |             |             |             |             |             | Sitting down         |             |             |             |             |             | Taking photo      |             |             |             |             |             |
| millisec | 80          | 160         | 320         | 400         | 560         | 1000        | 80          | 160         | 320         | 400         | 560         | 1000        | 80                   | 160         | 320         | 400         | 560         | 1000        | 80                | 160         | 320         | 400         | 560         | 1000        |
| [23]     | 0.60        | 0.86        | 1.24        | 1.30        | 1.58        | <b>2.26</b> | 0.44        | 0.74        | 1.19        | 1.40        | 1.57        | 2.03        | 0.51                 | 0.93        | 1.44        | 1.65        | 1.94        | 2.55        | 0.33              | 0.65        | 0.97        | 1.09        | 1.19        | 1.47        |
| TP-RNN   | <b>0.59</b> | <b>0.82</b> | <b>1.12</b> | <b>1.18</b> | <b>1.52</b> | 2.28        | <b>0.41</b> | <b>0.66</b> | <b>1.07</b> | <b>1.22</b> | <b>1.35</b> | <b>1.74</b> | <b>0.41</b>          | <b>0.79</b> | <b>1.13</b> | <b>1.27</b> | <b>1.47</b> | <b>1.93</b> | <b>0.26</b>       | <b>0.51</b> | <b>0.80</b> | <b>0.95</b> | <b>1.08</b> | <b>1.35</b> |
|          | Waiting     |             |             |             |             |             | Walking dog |             |             |             |             |             | Walking together     |             |             |             |             |             | Average of all 15 |             |             |             |             |             |
| millisec | 80          | 160         | 320         | 400         | 560         | 1000        | 80          | 160         | 320         | 400         | 560         | 1000        | 80                   | 160         | 320         | 400         | 560         | 1000        | 80                | 160         | 320         | 400         | 560         | 1000        |
| [23]     | 0.34        | 0.65        | 1.09        | <b>1.28</b> | <b>1.61</b> | <b>2.27</b> | 0.56        | 0.95        | 1.28        | 1.39        | <b>1.68</b> | <b>1.92</b> | 0.31                 | 0.61        | 0.84        | 0.89        | 1.00        | 1.43        | 0.43              | 0.75        | 1.11        | 1.24        | 1.42        | 1.83        |
| TP-RNN   | <b>0.30</b> | <b>0.60</b> | <b>1.09</b> | 1.31        | 1.71        | 2.46        | <b>0.53</b> | <b>0.93</b> | <b>1.24</b> | <b>1.38</b> | 1.73        | 1.98        | <b>0.23</b>          | <b>0.47</b> | <b>0.67</b> | <b>0.71</b> | <b>0.78</b> | <b>1.28</b> | <b>0.37</b>       | <b>0.66</b> | <b>0.99</b> | <b>1.11</b> | <b>1.30</b> | <b>1.71</b> |

outperforms other prior works in most of the cases. Another strong previous work in long-term forecasting is the Dropout-AutoEncoder model [16], which generates the best 1000ms prediction for the ‘Discussion’ action among all other models. Similar to short-term predictions, our proposed velocity-based model outperforms all the baseline and state-of-the-art methods, except for the Dropout-AutoEncoder model in 1000ms prediction with respect to only the ‘Discussion’ action. Note that our model conducts an action-agnostic forecasting and does not sacrifice the short-term or long-term predictions. Our results in comparison with other methods that are either trained for each action separately (like [15, 18]) or only target short- or long-term predictions (*e.g.*, [16, 23]) show better overall performance. As our models are trained

using all 15 actions in Human 3.6M [17], without any extra supervision from the action labels, we further evaluate the proposed method by reporting the average MAE for all time-points across all 15 action categories. In Tables 3 and 4, we compare our results with the current state-of-the-art model [23], which is the only previous research experimented on all 15 action classes. Table 3 shows the long-term forecasting results of the remaining 11 action categories, not included in Table 2. As can be seen, our proposed TP-RNN model performs better than [23] in most of the action categories.

Table 4 summarizes the short-term and long-term results from the current state-of-the-art model [23] (from the paper and the code, both when including action labels as inputs to the model or not), and the results from our proposed

Table 4: Comparison of average MAE across all 15 actions of Human 3.6 dataset with prior and baseline models (ablation study). In each column, the best results are typeset in bold and the second best are underlined. For ‘Residual [23]’, both short-term (from paper) and long-term (from code) results are reported. AA: Action-Agnostic.

|                            | AA | 80          | 160         | 320         | 400         | 560         | 1000        |
|----------------------------|----|-------------|-------------|-------------|-------------|-------------|-------------|
| Residual [23] (short)      | ✗  | <b>0.36</b> | <u>0.67</u> | 1.02        | 1.15        | -           | -           |
| Residual [23] (short)      | ✓  | 0.39        | 0.72        | 1.08        | 1.22        | -           | -           |
| Residual [23] (long)       | ✗  | 0.43        | 0.75        | 1.11        | 1.24        | 1.42        | 1.83        |
| Residual [23] (long)       | ✓  | 0.42        | 0.73        | 1.09        | 1.23        | 1.42        | 1.84        |
| Zero-velocity              | -  | 0.40        | 0.71        | 1.07        | 1.21        | 1.42        | 1.85        |
| Single Layer (Pose)        | ✓  | 0.49        | 0.83        | 1.20        | 1.34        | 1.53        | 1.92        |
| Single Layer (Vel.)        | ✓  | 0.39        | <u>0.67</u> | <u>1.00</u> | 1.13        | 1.32        | 1.73        |
| Stacked 2-Layer (Vel.)     | ✓  | 0.38        | <b>0.66</b> | 1.01        | 1.13        | 1.32        | 1.74        |
| Double-scale (Vel.)        | ✓  | <u>0.37</u> | <b>0.66</b> | <b>0.99</b> | <b>1.11</b> | <b>1.30</b> | 1.73        |
| Double-scale (Hier., Vel.) | ✓  | <u>0.37</u> | <b>0.66</b> | <u>1.00</u> | <u>1.12</u> | 1.32        | 1.76        |
| Double-scale (Phase, Vel.) | ✓  | <u>0.37</u> | <b>0.66</b> | <u>1.00</u> | <u>1.12</u> | <u>1.31</u> | <u>1.72</u> |
| TP-RNN (Ours)              | ✓  | <u>0.37</u> | <b>0.66</b> | <b>0.99</b> | <b>1.11</b> | <b>1.30</b> | <b>1.71</b> |

velocity-based models by showing the average MAE for all time-points across all 15 action categories. Our models outperform the Residual model, in both short-term and long-term prediction tasks. We use the average MAE metric to also conduct the ablation analysis by evaluating the difference between each of our model extensions with hierarchical and multi-scale architectures. We can see that basic multi-layer LSTM model (*i.e.*, Stacked 2-Layer (Vel.)) does not improve the overall performance compared with single layer LSTM model (*i.e.*, Single Layer (Vel.)). After we include the multi-scale idea in our models, we can see the improvement over the single layer model (*i.e.*, Single Layer (Vel.)) and the basic multi-layer single-scale model (*i.e.*, Stacked 2-Layer (Vel.)). For the long-term time-range, the model directly adapted from HM-RNN [12] (*i.e.*, Double-scale (Hier., Vel.)) does not improve the performance, because the original design of the HM-RNN model is for the natural language data with obvious boundaries in the sequence. We mitigate this limitation with our proposed TP-RNN model, which has multiple phases of LSTM on the higher levels for capturing the up-to-date larger scale temporal information. TP-RNN provides best quantitative results, as shown in Table 4.

In summary, Table 4 shows that the previous state-of-the-art (*i.e.*, Residual [23]) needs to compromise between short- and long-term forecasting accuracy. However, TP-RNN model performs better, especially for the long-term forecasting (1000ms in future) without sacrificing the short-term accuracy. In terms of numbers, our long-term forecasting improvement is  $\frac{1.83-1.71}{1.83} \propto 6.56\%$ , which is not negligible. Besides, Table 2 shows that for certain action categories with hierarchical multi-scale motions (e.g., ‘Walking’), long-term forecasting improvement is even more sig-

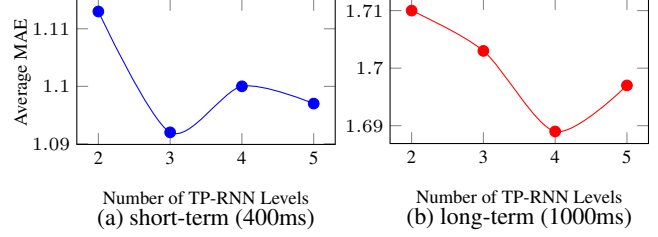


Figure 3: Average MAEs of short- and long-term forecasting using TP-RNN with different levels:  $M \in \{2, 3, 4, 5\}$ .

nificant:  $\frac{0.96-0.77}{0.96} \propto 19.79\%$ . To further test the significance of the improvements, we conduct a one-tailed paired t-test between our average results and those of [23]. The  $p$ -value of the test equals 0.0002, which by all conventional criteria the difference between the two sets of results is considered to be extremely *statistically significant*.

**Deeper Hierarchical Structure of TP-RNN:** In the previous subsections, we showed that even with the most basic architectural settings (*i.e.*,  $M = 2$  and  $K = 2$ ), TP-RNN already outperforms the state-of-the-art. In this section, we further experiment and analyze the effect of increasing the number of levels  $M$ . Fig. 3(b) shows the average MAE of the long-term forecasting (1000ms) results from TP-RNN with different numbers of levels:  $M \in \{2, 3, 4, 5\}$ . In general, increasing the number of levels of TP-RNN improves the long-term forecasting accuracy, which is in accordance with our hypothesis that higher hierarchical levels are able to better capture long-term human motion dynamics and thus improve the long-term forecasting accuracy. In addition, Fig 3(a) shows similar results for short-term forecasting (400ms), which also indicates that the performance improves slightly when increasing the number of levels. However, when we increase the number of TP-RNN levels to 4 for short-term or 5 for long-term, we see a decline in the performance. The reason is that the LSTM cell(s) at the 5<sup>th</sup> level only update once at every  $2^{5-1} = 16$  time-step, which is too long, given that we only predict the future 25 frames (*i.e.*, very few updates for  $M = 5$ ). Besides, with deeper hierarchical structures, TP-RNN has more trainable parameters and therefore is prone to overfitting (requires more data).

**Qualitative Evaluations and Visualization:** Fig. 4 shows the visualization of pose sequences for our method, in comparisons with the Residual method [23], for the sequences for the actions ‘Walking’ and ‘Smoking’. As it can be seen in the figure, our predictions (the middle row) are visually closer to the ground-truth (last row) compared to the state-of-the-art Residual method [23]. This also supports the quantitative results shown in Table 2, as our method steadily outperforms [23] for the ‘walking’ activity: TP-RNN’s MAE was 0.25 at 80ms while [23] led to an MAE of 0.32, similarly, ours was 0.75 and 0.77 in 560ms and 1000ms while [23] obtained 0.86

Table 5: Comparison with prior works on the Penn Action dataset in terms of PCK@0.05. Best results are typset in bold, and the second best are underlined. TP-RNN is tested with or without incorporating an initial pose velocity (details in the text).

| Future frame         | 1           | 2           | 3           | 4           | 5           | 6           | 7           | 8           | 9           | 10          | 11          | 12          | 13          | 14          | 15          | 16          |
|----------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Residual [23]        | 82.4        | 68.3        | 58.5        | 50.9        | 44.7        | 40.0        | 36.4        | 33.4        | 31.3        | 29.5        | 28.3        | 27.3        | 26.4        | 25.7        | 25.0        | 24.5        |
| 3D-PFNet [11]        | 79.2        | 60.0        | 49.0        | 43.9        | 41.5        | 40.3        | 39.8        | 39.7        | 40.1        | 40.5        | 41.1        | 41.6        | 42.3        | 42.9        | 43.2        | 43.3        |
| TP-RNN w/o init vel. | 82.3        | 68.9        | <u>61.5</u> | <u>56.9</u> | <u>53.9</u> | <u>51.7</u> | <u>50.0</u> | <u>48.5</u> | <u>47.3</u> | <u>46.2</u> | <u>45.6</u> | <u>45.0</u> | <u>44.6</u> | <u>44.3</u> | <u>44.1</u> | <u>43.9</u> |
| TP-RNN w/ init vel.  | <b>84.5</b> | <b>72.0</b> | <b>64.8</b> | <b>60.3</b> | <b>57.2</b> | <b>55.0</b> | <b>53.4</b> | <b>52.1</b> | <b>50.9</b> | <b>50.0</b> | <b>49.3</b> | <b>48.7</b> | <b>48.3</b> | <b>47.9</b> | <b>47.6</b> | <b>47.3</b> |

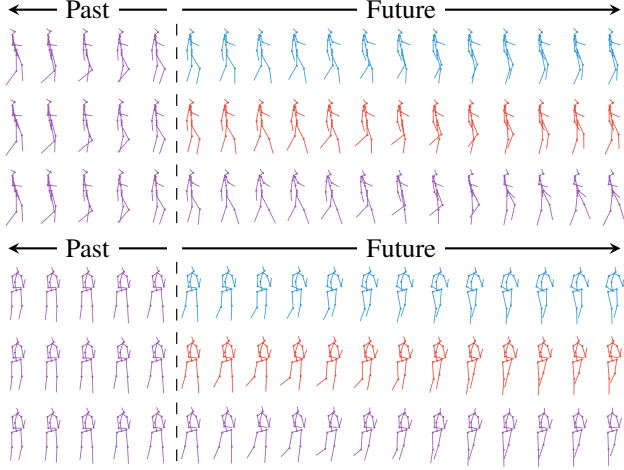


Figure 4: Visualization of long-term pose-forecasting (up to 1000ms) for the actions ‘Walking’ (top) and ‘Smoking’ (bottom), downsampled by a factor of 2 (*i.e.*, 13 forecasted poses out of 25 are visualized). The purple poses are ground-truth data, including past (in the left side) and future time frames (in the bottom). The blue and red poses are the predictions of [23] and our method (TP-RNN), respectively.

and 0.96, respectively. A similar conclusion can be made for predictions of the ‘smoking’ activity. Although, this activity has slight amounts of movement, still our method can capture better dynamics, if we look at the results closely. For instance, using our method, the distance of the subject’s hand from the torso and face are better predicted in long-term and leg movements are more precisely forecasted, especially in shorter-term predictions. More visualizations of the forecasted poses by TP-RNN were visualized in Fig. 1, in which one can simply observe how the patterns of poses change through time, in comparison with the ground-truth.

**Comparison on Penn Action Dataset:** We trained TP-RNN on Penn Action dataset [46] and, here, we compare its results with the previous state-of-the-art on this dataset, 3D-PFNet [11]. The input to our velocity-based TP-RNN is the pose in a single frame, and we set the initial velocity to 0 (denoted by TP-RNN w/o init vel.), in order to have a fair comparison with [11]. The model performance is evaluated using PCK@0.05 as in [11]. PCK calculates the percentage

of joint locations correctly predicted by the model. With the threshold 0.05, a joint location is counted as correctly predicted if the normalized distance between its predicted and ground-truth locations is less than 0.05. The results are shown in Table 5. Our model performs significantly better than 3D-PFNet [11] (a  $p$ -value of  $0.0419 < 0.05$  significance threshold), with the results shown in Table 5. For further comparison, we used the open-sourced code of [23] with the necessary modifications (the same setting as our TP-RNN), and experimented on the Penn Action dataset. Additionally, we further experiment by including the initial velocity in the input, to demonstrate that the importance of the velocity information. The additional initial velocity is estimated using the difference between the current pose and the previous pose, our TP-RNN model generates even better forecasting results, as shown in the last row in Table 5.

## 5. Conclusion

In this paper, inspired by the success of hierarchical multi-scale RNN (HM-RNN) frameworks in the natural language processing applications, we proposed a new model to encode different hierarchies in human dynamics at different time-scales. Our model trains a set of RNNs (as LSTM sequences) at each hierarchy with different time-scales. Within each level of the hierarchy, RNNs share their learnable weights, since they are all learning a same concept in a same time-scale (with different phases, *i.e.*, different starting points of the sequence). This dramatically decreases the number of parameters to be learned in the model, while involving as much data as possible to train the higher level RNNs. As a result, the lowest layer can learn the finest grained scale motion dynamics, and higher levels capture different characteristics of the dynamics each at a certain time-scale. Furthermore, we set up a more rigorous but realistic experimental settings by conducting an action-agnostic forecasting (*i.e.*, no usage of activity labels) and predicting both short- and long-term sequences simultaneously (unlike the previous works, which limited their settings). Despite these strict settings, our results on the Human 3.6M dataset and the Penn Action dataset show superior performance for TP-RNN to the baseline and state-of-the-art methods, in terms of both quantitative evaluations and qualitative visualizations.

**Acknowledgements** We thank Panasonic for their support.



## References

- [1] E. Adeli-Mosabbebi, R. Cabral, F. De la Torre, and M. Fathy. Multi-label discriminative weakly-supervised human activity recognition and localization. In *Asian Conference on Computer Vision*, pages 241–258. Springer, 2014.
- [2] M. S. Aliakbarian, F. S. Saleh, M. Salzmann, B. Fernando, L. Petersson, and L. Andersson. Encouraging lstms to anticipate actions very early. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 280–289, 2017.
- [3] M. Andriluka, S. Roth, and B. Schiele. Monocular 3d pose estimation and tracking by detection. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 623–630. IEEE, 2010.
- [4] M. Argyle. Non-verbal communication in human social interaction. Cambridge U. Press, 1972.
- [5] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.
- [6] M. M. Arzani, M. Fathy, H. Aghajan, A. A. Azirani, K. Raahemifar, and E. Adeli. Structured prediction with short/long-range dependencies for human activity recognition from depth skeleton data. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 560–567. IEEE, 2017.
- [7] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio. End-to-end attention-based large vocabulary speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 4945–4949. IEEE, 2016.
- [8] E. Barsoum, J. Kender, and Z. Liu. Hp-gan: Probabilistic 3d human motion prediction via gan. *arXiv preprint arXiv:1711.09561*, 2017.
- [9] J. Bütetage, M. J. Black, D. Kragic, and H. Kjellström. Deep representation learning for human motion prediction and classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 2017, 2017.
- [10] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, volume 1, page 7, 2017.
- [11] Y.-W. Chao, J. Yang, B. Price, S. Cohen, and J. Deng. Forecasting human dynamics from static images. In *CVPR*, 2017.
- [12] J. Chung, S. Ahn, and Y. Bengio. Hierarchical multiscale recurrent neural networks. In *ICLR*, 2017.
- [13] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.
- [14] S. El Hihi and Y. Bengio. Hierarchical recurrent neural networks for long-term dependencies. In *Advances in neural information processing systems*, pages 493–499, 1996.
- [15] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik. Recurrent network models for human dynamics. In *Computer Vision (ICCV), 2015 IEEE International Conference on*, pages 4346–4354. IEEE, 2015.
- [16] P. Ghosh, J. Song, E. Aksan, and O. Hilliges. Learning human motion models for long-term predictions. *arXiv preprint arXiv:1704.02827*, 2017.
- [17] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2014.
- [18] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5308–5317, 2016.
- [19] J. Koutnik, K. Greff, F. Gomez, and J. Schmidhuber. A clockwork rnn. In *International Conference on Machine Learning*, pages 1863–1871, 2014.
- [20] T. Lan, T.-C. Chen, and S. Savarese. A hierarchical representation for future action prediction. In *European Conference on Computer Vision*, pages 689–704. Springer, 2014.
- [21] P. Luc, N. Neverova, C. Couprie, J. Verbeek, and Y. LeCun. Predicting deeper into the future of semantic segmentation. In *of: ICCV 2017-International Conference on Computer Vision*, page 10, 2017.
- [22] R. Mahjourian, M. Wicke, and A. Angelova. Geometry-based next frame prediction from monocular video. In *Intelligent Vehicles Symposium (IV), 2017 IEEE*, pages 1700–1707. IEEE, 2017.
- [23] J. Martinez, M. J. Black, and J. Romero. On human motion prediction using recurrent neural networks. In *CVPR*, 2017.
- [24] J. Martinez, R. Hossain, J. Romero, and J. J. Little. A simple yet effective baseline for 3d human pose estimation. In *ICCV*, 2017.
- [25] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. In *ICLR*, 2016.
- [26] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt. Vnect: Real-time 3d human pose estimation with a single rgb camera. *ACM Transactions on Graphics (TOG)*, 36(4):44, 2017.
- [27] T. B. Moeslund and E. Granum. 3d human pose estimation using 2d-data and an alternative phase space representation. *Procedure Humans 2000*, 2000.
- [28] S. L. Pintea, J. C. van Gemert, and A. W. Smeulders. Déjà vu. In *European Conference on Computer Vision*, pages 172–187. Springer, 2014.
- [29] G. Rogez, J. Rihan, S. Ramalingam, C. Orrite, and P. H. Torr. Randomized trees for human pose detection. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.
- [30] G. Rogez, P. Weinzaepfel, and C. Schmid. Lcr-net: Localization-classification-regression for human pose. In *CVPR 2017-IEEE Conference on Computer Vision & Pattern Recognition*, 2017.
- [31] P. Schydlo, M. Rakovic, L. Jamone, and J. Santos-Victor. Anticipation in human-robot cooperation: A recurrent neural network approach for multiple action sequences prediction. *arXiv preprint arXiv:1802.10503*, 2018.
- [32] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1297–1304. Ieee, 2011.
- [33] L. Sigal. Human pose estimation. In *Computer Vision*, pages 362–370. Springer, 2014.

- [34] K. Soomro, H. Idrees, and M. Shah. Online localization and prediction of actions and interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.
- [35] D. Tome, C. Russell, and L. Agapito. Lifting from the deep: Convolutional 3d pose estimation from a single image. *CVPR 2017 Proceedings*, pages 2500–2509, 2017.
- [36] S. Toyer, A. Cherian, T. Han, and S. Gould. Human pose forecasting via deep markov models. *arXiv preprint arXiv:1707.09240*, 2017.
- [37] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *Advances In Neural Information Processing Systems*, pages 613–621, 2016.
- [38] J. Walker, C. Doersch, A. Gupta, and M. Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *European Conference on Computer Vision*, pages 835–851. Springer, 2016.
- [39] J. Walker, A. Gupta, and M. Hebert. Dense optical flow prediction from a static image. In *Computer Vision (ICCV), 2015 IEEE International Conference on*, pages 2443–2451. IEEE, 2015.
- [40] J. Walker, K. Marino, A. Gupta, and M. Hebert. The pose knows: Video forecasting by generating pose futures. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3352–3361. IEEE, 2017.
- [41] D. Wang and E. Nyberg. A long short-term memory model for answer sentence selection in question answering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 707–712, 2015.
- [42] J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):283–298, 2008.
- [43] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016.
- [44] D. Wu and L. Shao. Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 724–731, 2014.
- [45] M. Ye, X. Wang, R. Yang, L. Ren, and M. Pollefeys. Accurate 3d pose estimation from a single depth image. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 731–738. IEEE, 2011.
- [46] W. Zhang, M. Zhu, and K. Derpanis. From actemes to action: A strongly-supervised representation for detailed action understanding. In *ICCV*, 2013.