# Attention Mechanism Exploits Temporal Contexts: Real-time 3D Human Pose Reconstruction

Ruixu Liu[1], Ju Shen[1], He Wang[1], Chen Chen[2], Sen-ching Cheung[3], Vijayan Asari[1]

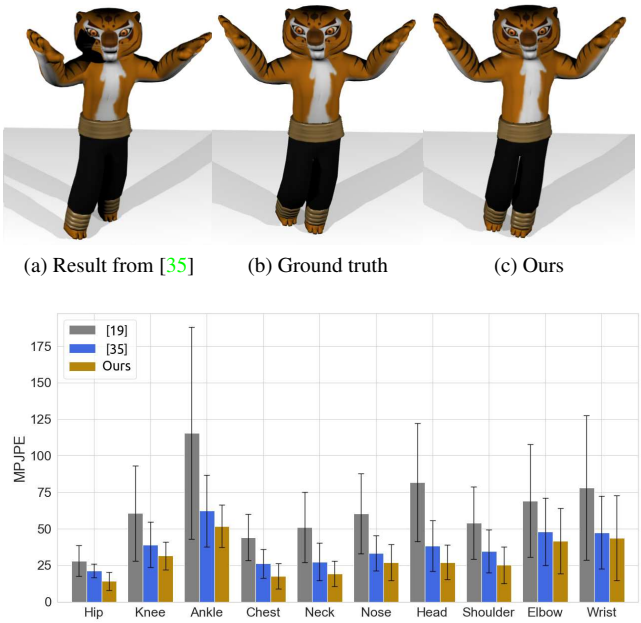
[1]University of Dayton, [2]University of North Carolina at Charlotte, [3]University of Kentucky
{liur05, jshen1, hwang6, vasari1}@udayton.edu, chen.chen@uncc.edu, sccheung@ieee.org

## Abstract

*We propose a novel attention-based framework for 3D human pose estimation from a monocular video. Despite the general success of end-to-end deep learning paradigms, our approach is based on two key observations: (1) temporal incoherence and jitter are often yielded from a single frame prediction; (2) error rate can be remarkably reduced by increasing the receptive field in a video. Therefore, we design an attentional mechanism to adaptively identify significant frames and tensor outputs from each deep neural net layer, leading to a more optimal estimation. To achieve large temporal receptive fields, multi-scale dilated convolutions are employed to model long-range dependencies among frames. The architecture is straightforward to implement and can be flexibly adopted for real-time applications. Any off-the-shelf 2D pose estimation system, e.g. Mocap libraries, can be easily integrated in an ad-hoc fashion. We both quantitatively and qualitatively evaluate our method on various standard benchmark datasets (e.g. Human3.6M, HumanEva). Our method considerably outperforms all the state-of-the-art algorithms up to* **8%** *error reduction (average mean per joint position error: 34.7) as compared to the best-reported results. Code is available at: (https://github.com/lrxjason/Attention3DHumanPose)*

## 1. Introduction

Articulated 3D human pose estimation is a classic vision task enabling numerous applications from activity recognition to human-robot interaction. Traditional approaches often use specialized devices under highly controlled environments, such as multi-view capture [1], marker systems [26] and multi-modal sensing [32], which requires a laborious setup process that limits their practical uses. This work focuses on 3D pose estimation from an arbitrary monocular video, which is challenging due to the high-dimensional variability and nonlinearity of human dynamics. Recent efforts of using deep architectures have significantly advanced the state-of-the-art in 3D pose reasoning [41, 29]. The end-to-end learning process alleviates the need of using tailor-made features or spatial constraints, thereby minimizing the characteristic errors such as double-counting image evidence [15].




(a) Result from [35] (b) Ground truth (c) Ours

Figure 1: Comparison results: Top: side-by-side views of motion retargeting results on a 3D avatar; the source is from frame 857 of *walking S9* and frame 475 of *posing S9* in Human3.6M. Bottom: the average joint error comparison across all the frames of the video walking S9 [19, 35].

In this work, we aim to utilize an attention model to further improve the accuracy among existing deep networks while preserving natural temporal coherence in videos. The

concept of "attention" is to learn optimized global alignment between pairwise data and has gained recent success in the integration with deep networks for processing mono/multi-modal data, such as text-to-speech matching [12] or neural machine translation [3]. To the best of our knowledge, our work is the first to use the attention mechanism in the domain of 3D pose estimation to selectively identify important tensor through-puts across neural net layers to reach an optimal inference.

While vast and powerful deep models on 3D pose prediction are emerging (from convolutional neural network (CNN) [34, 40, 22] to generative adversarial networks (GAN) [43, 10]), many of these approaches focus on a single image inference, which is inclined to jittery motion or inexact body configuration. To resolve this, temporal information is taken into account for better motion consistency. Existing works can be generally classified into two categories: *direct 3D estimation* and *2D-to-3D estimation* [50, 9]. The former explores the possibility of jointly extracting both 2D and 3D poses in a holistic manner [34, 42]; while the latter decouples the estimation into two steps: 2D body part detection and 3D correspondence inference [8, 5, 50]. We refer readers to the recent survey for more details of their respective advantages [27].

Our approach falls under the category of *2D-to-3D estimation* with two key contributions: (a) developing a systematic approach to design and train of attention models for 3D pose estimation and (b) learning implicit dependencies in large temporal receptive fields using multi-scale dilated convolutions. Experimental evaluations show that the resulting system can reach almost the same level of estimation accuracy under both causal or non-causal conditions, making it very attractive for real-time or consumer-level applications. To date, state-of-the-art results on video-based *2D-to-3D estimation* can be achieved by a semi-supervised approach [35] or a layer normalized LSTM approach [19]. Our model can further improve the performance in both quantitative accuracy and qualitative evaluation. Figure 1 shows an example result from *Human3.6M* measured by the *Mean Per Joint Position Error (MPJPE)*. To visually demonstrate the significance of the improvement, animation retargeting is applied to a 3D avatar by synthesizing the captured motion from the same frame of the *Walking S9* and *posing S9* sequences. From the side-by-side comparisons, one can easily see the differences of the rendered results against the ground truth. Specifically, the shadows of the legs and the right hand are rendered differently due to the erroneous pose estimated, while ours stay more aligned with the ground truth. The histogram on the bottom demonstrates the MPJPE error reduction on individual joints. More extensive evaluation can be found in our supplementary materials.

## 2. Related Works

Articulated pose estimation from a video has been studied for decades. Early works relied on graphical or restrictive models to account for the high degree of freedom and dependencies among body parts, such as tree-structures [2, 1, 44] or pictorial structures [2]. These methods often introduced a large number of parameters that required careful and manual tuning using techniques such as piecewise approximation. With the rise of convolutional neural networks (CNNs) [34, 38], automated feature learning disentangles the dependencies among output variables and surpasses the performance of tailor-made solvers. For example, Tekin *et al*. trained an auto-encoder to project 3D joints to a high dimensional space to enforce structural constraints [40]. Park *et al*. estimated the 3D pose by propagating 2D classification results to 3D pose regressors inside a neural network [33]. A kinematic object model was introduced to guarantee the geometric validity of the estimated body parts [49]. A comprehensive list on CNNs-based systems can be found in the survey [38].

Our contribution to this rich body of works lies in the introduction of attention mechanism that can further improve the estimation accuracy on traditional convolutional networks. Prior work on attention in deep learning (DL) mostly addresses long short-term memory networks (LSTMs) [18]. For example, a LSTM encodes *context* within a sentence to form attention-based word representations that boost the word-alignment between two sentences [36]. A similar attentional mechanism was successfully applied to improve the task of neural machine translation by jointly translating and aligning words [3]. Given the success in the language domain, we utilize the attention model for visual data computing through training a temporal convolutional network (TCN) [45].

Compared to LSTMs, TCNs have the advantage of efficient memory usage without storing a large number of parameters introduced by the gates of LSTMs [31, 4]. In addition, TCNs enable parallel processing on the input frames instead of sequentially loading them into memory [19], where an estimation failure on one frame might affect the subsequent ones. Our work bears some similarity to the semi-supervised approach that uses a voting mechanism to select important frames [35]. But ours has three distinct features: first, instead of selectively choosing a subset of frames for estimation, our approach systematically assign a weight distribution to frames, all of which might contribute to the inference. Furthermore, our attention model enables automated weight assignment to all the network tensors and their internal channels that significantly improve the accuracy. Last but not least, our dilation model aims at enhancing the temporal consinstency with large receptive field, while the semi-supervised approach focuses on speeding up the computation by reusing pre-processed frames [35].

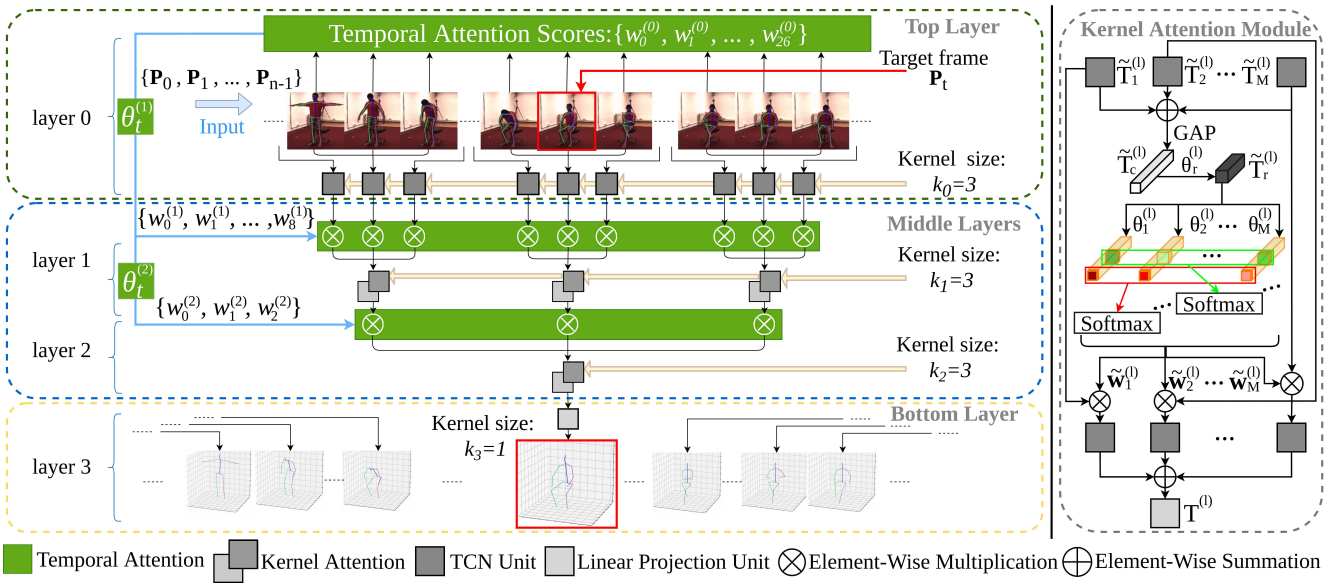



Figure 2: Left: An example of 4-layers architecture for attention-based temporal convolutional neural network. In this example, all the kernel sizes are 3. In practice, different layers can have different kernel sizes. Right: The detailed configuration of Kenrnel Attention Module.

## 3. The Attention-based Approach

### 3.1. Network Design

Figure 2 (left) depicts the overall architecture of our attention-based neural network. It takes a sequence of $n$ frames with 2D joint positions as the input and outputs the estimated 3D pose for the target frame as labeled. The framework involves two types of processing modules: the *Temporal Attention* module (indicated by the long green bars) and the *Kernel Attention* module (indicated by the gray squares). The kernel attention module can be further categorized as TCN Units (in dark grey color) and Linear Projection Units (in light grey color) [17]. By viewing the graphical model vertically from the top, one can notice the two attention modules distribute in an interlacing pattern that a row of kernel attention modules situate right below a temporal attention module. We regard these two adjacent modules as one *layer*, which has the same notion as a neural net layer. According to the functionalities, the layers can be grouped as *top layer*, *middle layers*, and *bottom layer*. Note that the top layer only has TCN units for the kernel module, while the bottom layer only has a linear projection unit to deliver the result. It is also worth mentioning that the number of middle layers can be varied depending on the receptive field setting, which will be discussed in section 5.3.

### 3.2. Temporal Attention

The goal of the temporal attention module is to provide a contribution metric for the output tensors. Each attention module produces a set of scalars, $\{\omega_0^{(l)}, \omega_1^{(l)}, \dots\}$, weighing the significance of different tensors within a layer:

$$\mathbf{W}^{(l)} \otimes \mathbf{T}^{(l)} \triangleq \left\{\omega_0^{(l)} \otimes \mathcal{T}_0^{(l)}, \dots, \omega_{\lambda_l-1}^{(l)} \otimes \mathcal{T}_{\lambda_l-1}^{(l)}\right\} \quad (1)$$

where $l$ and $\lambda_l$ indicate the layer index and the number of tensors output from the $l^{(th)}$ layer. We use $\mathcal{T}_u^{(l)}$ to denote the $u^{th}$ tensor output from the $l^{th}$ layer. The bold format of $\mathbf{W} \otimes \mathbf{T}$ is a compacted vector representation used in Algorithm 1. Note for the top layer, the input to the TCN units is just the 2D joints. The choice for computing their attention scores can be flexible. A commonly used scheme is the multilayer perceptron strategy for optimal feature set selection [37]. Empirically, we achieve desirable result by simply computing the *normalized cross-correlation (ncc)* that measures the positive cosine similarity between $\mathbf{P}_i$ and $\mathbf{P}_t$ on their 2D joint positions [46]:

$$\mathbf{W}^{(0)} = [ncc(\mathbf{P}_0, \mathbf{P}_t), \dots, ncc(\mathbf{P}_{n-1}, \mathbf{P}_t)]^T \quad (2)$$

where $\mathbf{P}_0, \dots, \mathbf{P}_{n-1}$ are the 2D joint positions. $t$ indicates the target frame index. The output $\mathbf{W}^{(0)}$ is forwarded to the attention matrix $\boldsymbol{\theta_t}^{(l)}$ to produce tensor weights for the subsequent layers.

$$\mathbf{W}^{(l)} = sig\left(\boldsymbol{\theta_t}^{(l)T}\mathbf{W}^{(l-1)}\right), \text{ for } l \in [1, L-2] \quad (3)$$

where $sig(\cdot)$ is the sigmoid activation function. We require the dimension of $\boldsymbol{\theta_t}^{(l)} \in \mathcal{R}^{F' \times F}$ matching the number of output tensors between layers $l-1$ and $l$, s.t. $F' = \lambda_{l-1}$ and $F = \lambda_l$.

### 3.3. Kernel Attention

Similar to the temporal attention that determines a tensor weight distribution $\mathbf{W}^{(l)}$ within layer $l$, the kernel attention module assigns a channel weight distribution within a tensor, denoted as $\widetilde{\boldsymbol{W}}^{(l)}$. Figure 2 (right) depicts the steps on how an updated tensor $\mathbf{T}_{final}^{(l)}$ is generated through the weight adjustment. Given an input tensor $\mathbf{T}^{(l)} \in \mathcal{R}^{C\times F}$, we generate $M$ new tensors $\widetilde{T}_m^{(l)}$ using $M$ TCN units with different dilation rates. These $M$ tensors are fused together through element-wise summation: $\widetilde{\mathbf{T}}^{(l)} = \sum_{m=1}^{M} \widetilde{T}_m^{(l)}$, which is fed into a global average pooling layer (GAP) to generate channel-wise statistics $\widetilde{\mathcal{T}}_c^{(l)} \in \mathcal{R}^{C\times 1}$. The channel number $C$ is acquired through a TCN unit as discussed in the ablation study. The output $\widetilde{\mathcal{T}}_c^{(l)}$ is forwarded to a fully connected layer to learn the relationship among features of different kernel sizes: $\widetilde{\mathcal{T}}_r^{(l)} = \boldsymbol{\theta_r}^{(l)} \widetilde{\mathcal{T}}_c^{(l)}$. The role of matrix $\boldsymbol{\theta_r}^{(l)} \in \mathcal{R}^{r\times C}$ is to reduce the channel dimension to $r$. Guided by the compacted feature descriptor $\widetilde{\mathcal{T}}_r^{(l)}$, $M$ vectors are generated (indicated by the yellow cuboids) through a second fully connected layer across channels. Their kernel attention weights are computed by a softmax function:

$$\widetilde{\boldsymbol{W}}^{(l)} \triangleq \left\{ \widetilde{W}_1^{(l)}, ..., \widetilde{W}_M^{(l)} \,\middle|\, \widetilde{W}_m^{(l)} = \frac{e^{\boldsymbol{\theta_m}^{(l)} \widetilde{\mathcal{T}}_r^{(l)}}}{\sum_{m=1}^{M} e^{\boldsymbol{\theta_m}^{(l)} \widetilde{\mathcal{T}}_r^{(l)}}} \right\} \tag{4}$$

where $\boldsymbol{\theta_m}^{(l)} \in \mathcal{R}^{C\times r}$ are the kernel attention parameters and $\sum_{m=1}^{M} W_m^{(l)} = 1$. Based on the weight distribution, we finally obtain the output tensor:

$$\mathbf{T}_{final}^{(l)} \triangleq \sum_{m=1}^{M} \widetilde{W}_m^{(l)} \otimes \widetilde{T}_m^{(l)} \tag{5}$$

The channel update procedure can be further decomposed as:

$$\widetilde{W}_m^{(l)} \otimes \widetilde{T}_m^{(l)} = \left\{ \widetilde{\omega}_1^{(l)} \otimes \widetilde{\mathcal{T}}_1^{(l)}, \ldots, \widetilde{\omega}_C^{(l)} \otimes \widetilde{\mathcal{T}}_C^{(l)} \right\} \tag{6}$$

This shares the same format as the tensor distribution process (equation 1) in the temporal attention module but focuses on the channel distribution. The temporal attention parameters $\boldsymbol{\theta_t}^{(l)}$ and kernel attention parameters $\boldsymbol{\theta_r}^{(l)}$, $\boldsymbol{\theta_m}^{(l)}$ for $l \in [1, L-2]$ are learned through mini-batch stochastic gradient descent (SGD) in the same manner as the TCN unit training [6].

## 4. Integration with Dilated Convolutions

For the proposed attention model, a large receptive field is crucial to learn long range temporal relationships across frames, thereby enhancing the estimation consistency. However, with more frames feeding into the network, the number of neural layers increases together with more training parameters. To avoid vanishing gradients or other superfluous layers problems [27], we devise a multi-scale dilation (MDC) strategy by integrating dilated convolutions.

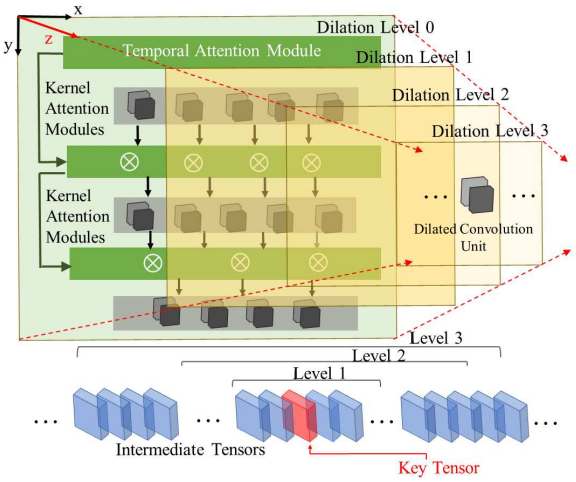



Figure 3: The model of temporal dilated convolution network. As the level index increases, the receptive field over frames (layer index = 0) or tensors (layer index $\geq$ 0) increases.

Figure 3 shows our dilated network architecture. For visualization purpose, we project the network into an $xyz$ space. The $xy$ plane has the same configuration as the network in Figure 2, with the combination of temporal and kernel attention modules along the $x$ direction, and layers layout along the $y$ direction. As an extension, we place the dilated convolution units (DCUs) along the $z$ direction. Terminologically, this $z$-axis is labeled as *level*s to differ from the *layer* concept along the $y$ direction. As the level index increases, the receptive field grows with increasing dilation size while reducing the number of DCUs.

Algorithm 1 describes the data flow on how these DCUs interact with each other. For notation simplicity, we use $\mathbf{U}_v^{(l)}$ to denote a DCU from layer $l$ and level $v$. With the extra dimension introduced by the dilation levels, the tensor's weights from the attention module in equation (1) are extended to three dimensional. We format them as a set of matrices: $\{\bar{\mathbf{W}}^{(0)}, \ldots, \bar{\mathbf{W}}^{(L-2)}\}$. Accordingly, the pre-learned attention parameters in equation (3) are upgraded to a tensor format $\{\hat{\boldsymbol{\theta}}_{\boldsymbol{t}}^{(1)}, \ldots, \hat{\boldsymbol{\theta}}_{\boldsymbol{t}}^{(L-2)}\}$. Lines 4∼5 of the Algorithm 1 provide the details about the dimension of a convolution unit, i.e. $kernel \times dilation \times stride$. For tensor product convenience, we impose the following dimension constraints to $\mathbf{U}_v^{(l)}$:

– The dilation size of unit $\mathbf{U}_v^{(l)}$ equals to the kernel size of the unit $\mathbf{U}_0^{(l+1)}$: $d_v^{(l)} := k_0^{(l+1)}$. In other words, the

**Algorithm 1:** Multi-scale Dilation Configuration

**input:** Number of layers: $L$
kernel sizes: $\{k_0, k_1, \ldots, k_{L-2}, 1\}$
2D joints: $\{\mathbf{P}_0, \mathbf{P}_1, \ldots, \mathbf{P}_{n-1}\}$

**Result:** configure the input/output for each $\mathbf{U}_v^{(l)}$

1 $V \coloneqq L - 2$ ; // level size
2 **for** $l \leftarrow 0$ **to** $L - 2$ **do**
3  **for** $v \leftarrow 0$ **to** $V - 1$ **do**
4  $d_v^{(l)} \coloneqq k_0^{(l+1)}$ ; // dilation size for $\mathbf{U}_v^{(l)}$
5  $s_v^{(l)} \coloneqq k_v^{(l)} \times d_v^{(l)}$ ; // stride size
6  $U_v^{(l)} = DCU(d_v^{(l)}, s_v^{(l)})$ **if** $l$ = *0* **then**
7  $\{\mathbf{P}_1, \ldots, \mathbf{P}_n\} \rightarrowtail \mathbf{U}_v^{(0)}$; // input
8  $\mathbf{U}_v^{(0)} \Rightarrow \mathbf{T}_v^{(0)}$ ; // output
9  **else**
10  $\bar{\mathbf{W}}_v^{(l)} = sig(\hat{\boldsymbol{\theta}_t}^{(l)T} \bar{\mathbf{W}}^{(l-1)})$
11  **if** $v = 0$ **then**
12  $i_m \coloneqq l - 1$ ; // max level index
13  $\{\bar{\mathbf{W}}_0^{(l-1)} \otimes \mathbf{T}_0^{(l-1)} \oplus \bar{\mathbf{W}}_1^{(l-1)} \otimes \mathbf{T}_1^{(l-2)} \oplus \cdots \oplus \bar{\mathbf{W}}_{i_m}^{(l-1)} \otimes \mathbf{T}_{i_m}^{(0)}\} \rightarrowtail \mathbf{U}_v^{(l)}$ ; // $\oplus$ is element-wise add
14  $\mathbf{U}_v^{(l)} \Rightarrow \mathbf{T}_v^{(l)}$ ;
15  **else**
16  $\bar{\mathbf{W}}_{i_m}^{(l-1)} \otimes \mathbf{T}_0^{(l-1)} \rightarrowtail \mathbf{U}_v^{(l)}$;
17  $\mathbf{U}_v^{(l)} \Rightarrow \mathbf{T}_v^{(l)}$ ;
18  **end**
19  **end**
20  **end**
21 **end**

dilation size of all the units from layer $l$ is defined by the kernel size of the $0^{th}$ unit of the next layer $l + 1$.

– The stride size of $\mathbf{U}_v^{(l)}$ equals to the product of its corresponding kernel and dilation sizes: $s_v^{(l)} \coloneqq k_v^{(l)} \times d_v^{(l)}$.

Lines 6 - 18 configure the input (denoted by "$\rightarrowtail$") and output (denoted by "$\Rightarrow$") data flows for the unit $\mathbf{U}_v^{(l)}$. For the input flow, we consider two cases according to the layer indices: $l = 0$ and $l \geq 1$. All the units from layer $l = 0$ share the same $n$ video frames as the input. For all the units from subsequent layers ($l \geq 1$), their input tensors are from:

$$input(\mathbf{U}_v^{(l)}) \triangleq \begin{cases} \{\mathbf{T}_0^{(l-1)}, \mathbf{T}_1^{(l-2)}, \ldots, \mathbf{T}_V^{(0)}\} & \text{if } v = 0; \\ \mathbf{T}_0^{(l-1)} & \text{otherwise.} \end{cases} \quad (7)$$

where $\mathbf{T}_v^{(l-1)}$ are the output tensors from the previous layer. Element-wise multiplication is applied to these input tensors with their weights $\bar{\mathbf{W}}_v^{(l-1)}$, as described in line 13.

## 5. Experiments

We have implemented the proposed approach in native Python without parallel optimization. The test system runs on a single NVIDIA TITAN RTX GPU. For real-time inference, it can reach 3000 FPS, approximately 0.3 milliseconds to process a video frame. For training and testing, we have built three prototypes $n = 27$, $n = 81$, and $n = 243$, where $n$ is the receptive field on input frames. The details about $n$'s selection is discussed in the ablation study section 5.3. All the prototypes present similar convergence rates in training and testing, as shown in Figure 4. We train our model using a ranger optimizer for 80 epochs with an initial learning rate of 1e-3, followed by a learning rate decay with cosine annealing decrease to 1e-5 [47, 24]. Data augmentation is applied to both the training and testing data by horizontally flipping poses. We also set the batch size, dropout rate, and activation function to 1024, 0.2, and Mish, respectively [35, 28].

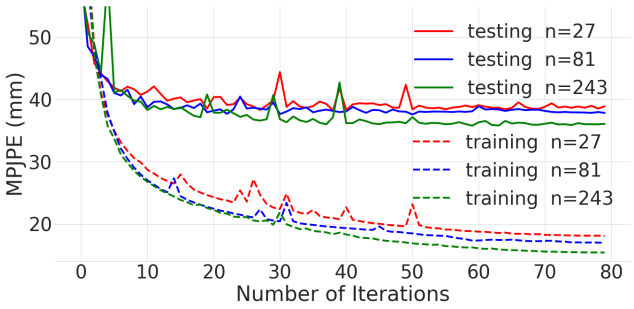



Figure 4: Convergence and accuracy performance for training and testing on the three prototypes.

### 5.1. Datasets and Evaluation Protocols

Our training images are from two public datasets: Human3.6M [7] and HumanEva [39], following the same training and validation policy as existing works [27, 43, 19, 35]. Specifically, the subjects S1, S5, S6, S7, and S8 from Human3.6M are used for training, and S9 and S11 are applied for testing. In the same manner, we conduct training/testing on the HumanEva dataset with the "Walk" and "Jog" actions performed by subjects S1, S2, and S3. For both datasets, we use the standard evaluation metrics (MPJPE and P-MPJPE) to measure the offset between the estimated result and ground-truth (GT) relative to the root node in millimeters [7]. Two protocols are involved in the experiment: $Protocol\#1$ computes the mean Euclidean distance for all the joints after aligning the root joints (i.e. *pelvis*) between the predicted and ground-truth poses, referred as MPJPE [14, 21, 34, 25]. $Protocol\#2$ applies an additional similarity transformation (Procrustes analysis) [20] to the predicted pose as an enhancement, referred as P-MPJPE

| Method | Dir. | Disc. | Eat | Greet | Phone | Photo | Pose | Pur. | Sit | SitD. | Smoke | Wait | WalkD. | Walk | WalkT. | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Martinez et al. ICCV'17 [27] | 51.8 | 56.2 | 58.1 | 59.0 | 69.5 | 78.4 | 55.2 | 58.1 | 74.0 | 94.6 | 62.3 | 59.1 | 65.1 | 49.5 | 52.4 | 62.9 |
| Fang et al. AAAI'18 [14] | 50.1 | 54.3 | 57.0 | 57.1 | 66.6 | 73.3 | 53.4 | 55.7 | 72.8 | 88.6 | 60.3 | 57.7 | 62.7 | 47.5 | 50.6 | 60.4 |
| Yang et al. CVPR'18 [43] | 51.5 | 58.9 | 50.4 | 57.0 | 62.1 | 65.4 | 49.8 | 52.7 | 69.2 | 85.2 | 57.4 | 58.4 | 43.6 | 60.1 | 47.7 | 58.6 |
| Pavlakos et al. CVPR'18 [34] | 48.5 | 54.4 | 54.4 | 52.0 | 59.4 | 65.3 | 49.9 | 52.9 | 65.8 | 71.1 | 56.6 | 52.9 | 60.9 | 44.7 | 47.8 | 56.2 |
| Luvizon et al. CVPR'18 [25] | 49.2 | 51.6 | 47.6 | 50.5 | 51.8 | 60.3 | 48.5 | 51.7 | 61.5 | 70.9 | 53.7 | 48.9 | 57.9 | 44.4 | 48.9 | 53.2 |
| Hossain et al. ECCV'18 [19] | 48.4 | 50.7 | 57.2 | 55.2 | 63.1 | 72.6 | 53.0 | 51.7 | 66.1 | 80.9 | 59.0 | 57.3 | 62.4 | 46.6 | 49.6 | 58.3 |
| Lee et al. ECCV'18 [21] | **40.2** | 49.2 | 47.8 | 52.6 | 50.1 | 75.0 | 50.2 | 43.0 | **55.8** | 73.9 | 54.1 | 55.6 | 58.2 | 43.3 | 43.3 | 52.8 |
| Dabral et al. ECCV'18 [13] | 44.8 | 50.4 | 44.7 | 49.0 | 52.9 | 61.4 | 43.5 | 45.5 | 63.1 | 87.3 | 51.7 | 48.5 | 52.2 | 37.6 | 41.9 | 52.1 |
| Zhao et al. CVPR'19 [48] | 47.3 | 60.7 | 51.4 | 60.5 | 61.1 | 49.9 | 47.3 | 68.1 | 86.2 | 55.0 | 67.8 | 61.0 | **42.1** | 60.6 | 45.3 | 57.6 |
| Pavllo et al. CVPR'19 [35] | 45.2 | 46.7 | 43.3 | 45.6 | 48.1 | 55.1 | 44.6 | 44.3 | 57.3 | 65.8 | 47.1 | 44.0 | 49.0 | 32.8 | 33.9 | 46.8 |
| Ours (n=243 CPN causal) | 42.3 | 46.3 | 41.4 | 46.9 | 50.1 | 56.2 | 45.1 | 44.1 | 58.0 | 65.0 | 48.4 | 44.5 | 47.1 | 32.5 | 33.2 | 46.7 |
| Ours (n=243 CPN) | 41.8 | **44.8** | **41.1** | **44.9** | **47.4** | **54.1** | **43.4** | **42.2** | 56.2 | **63.6** | **45.3** | **43.5** | 45.3 | **31.3** | **32.2** | **45.1** |
| Martinez et al. ICCV'17 [27] | 37.7 | 44.4 | 40.3 | 42.1 | 48.2 | 54.9 | 44.4 | 42.1 | 54.6 | 58.0 | 45.1 | 46.4 | 47.6 | 36.4 | 40.4 | 45.5 |
| Hossain et al. ECCV'18 [19] | 35.2 | 40.8 | 37.2 | 37.4 | 43.2 | 44.0 | **38.9** | **35.6** | 42.3 | 44.6 | 39.7 | 39.7 | 40.2 | 32.8 | 35.5 | 39.2 |
| Lee et al. ECCV'18 [21] | **32.1** | **36.6** | 34.4 | 37.8 | 44.5 | 49.9 | 40.9 | 36.2 | 44.1 | 45.6 | 35.3 | 35.9 | 37.6 | 30.3 | 35.5 | 38.4 |
| Zhao et al. CVPR'19 [48] | 37.8 | 49.4 | 37.6 | 40.9 | 45.1 | 41.4 | 40.1 | 48.3 | 50.1 | 42.2 | 53.5 | 44.3 | 40.5 | 47.3 | 39.0 | 43.8 |
| Pavllo et al. CVPR'19 [35] | 35.2 | 40.2 | **32.7** | 35.7 | 38.2 | 45.5 | 40.6 | 36.1 | 48.8 | 47.3 | 37.8 | 39.7 | 38.7 | 27.8 | 29.5 | 37.8 |
| Ours (n=243 GT) | 34.5 | 37.1 | 33.6 | **34.2** | **32.9** | **37.1** | 39.6 | 35.8 | **40.7** | **41.4** | **33.0** | **33.8** | **33.0** | **26.6** | **26.9** | **34.7** |

Table 1: $Protocol\#1$ with MPJPE (mm): Reconstruction error on Human3.6M. Top-table: input 2D joints are acquired by detection. Bottom-table: input 2D joints with ground-truth. (CPN) - cascaded pyramid network; (GT) - ground-truth.

| Method | Dir. | Disc. | Eat | Greet | Phone | Photo | Pose | Pur. | Sit | SitD. | Smoke | Wait | WalkD. | Walk | WalkT. | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Martinez et al. ICCV'17 [27] | 39.5 | 43.2 | 46.4 | 47.0 | 51.0 | 56.0 | 41.4 | 40.6 | 56.5 | 69.4 | 49.2 | 45.0 | 49.5 | 38.0 | 43.1 | 47.7 |
| Fang et al. AAAI'18 [14] | 38.2 | 41.7 | 43.7 | 44.9 | 48.5 | 55.3 | 40.2 | 38.2 | 54.5 | 64.4 | 47.2 | 44.3 | 47.3 | 36.7 | 41.7 | 45.7 |
| Hossain et al. ECCV'18 [19] | 35.7 | 39.3 | 44.6 | 43.0 | 47.2 | 54.0 | 38.3 | 37.5 | 51.6 | 61.3 | 46.5 | 41.4 | 47.3 | 34.2 | 39.4 | 44.1 |
| Pavlakos et al. CVPR'18 [34] | 34.7 | 39.8 | 41.8 | 38.6 | 42.5 | 47.5 | 38.0 | 36.6 | 50.7 | 56.8 | 42.6 | 39.6 | 43.9 | 32.1 | 36.5 | 41.8 |
| Yang et al. CVPR'18 [43] | **26.9** | 30.9 | 36.3 | 39.9 | 43.9 | 47.4 | **28.8** | **29.4** | **36.9** | 58.4 | 41.5 | **30.5** | 29.5 | 42.5 | 32.2 | 37.7 |
| Dabral et al. ECCV'18 [13] | 28.0 | **30.7** | 39.1 | **34.4** | 37.1 | **28.9** | 31.2 | 39.3 | 60.6 | **39.3** | 44.8 | 31.1 | **25.3** | 37.8 | 28.4 | 36.3 |
| Pavllo et al. CVPR'19 [35] | 34.1 | 36.1 | 34.4 | 37.2 | 36.4 | 42.2 | 34.4 | 33.6 | 45.0 | 52.5 | 37.4 | 33.8 | 37.8 | 25.6 | 27.3 | 36.5 |
| Ours (n=243 CPN) | 32.3 | 35.2 | **33.3** | 35.8 | **35.9** | 41.5 | 33.2 | 32.7 | 44.6 | 50.9 | **37.0** | 32.4 | 37.0 | **25.2** | **27.2** | **35.6** |

Table 2: $Protocol\#2$ with P-MPJPE (mm): Reconstruction error on Human3.6M with similarity transformation.

[27, 19, 43, 35]. Compared to $Protocol\#1$, this protocol is more robust to individual joint prediction failure. Another commonly used protocol (N-MPJPE) is to apply a scale alignment to the predicted pose. Compared to $Protocol\#2$, this protocol involves a relatively less degree of transformation, resulting in a smaller error range than $Protocol\#2$. Thus it should be sufficient to combine $Protocols\#1\&\#2$ for the accuracy analysis.

## 5.2. Comparison with State-of-the-Art

We compare our approach with state-of-the-art techniques on the two datasets Human3.6M and HumanEva, as shown in Tables 1-3. The best and second best results are highlighted in bold and underline formats respectively. The last column of each table shows the average performance on all the testing sets. Our approach achieves the minimum errors with 45.1mm in MPJPE and 35.6mm in P-MPJPE. In particular, under $Protocol\#1$, our model reduces the best reported error rate of MPJPE [35] by approximate $8\%$.

**2D Detection:** a number of widely adopted 2D detectors were investigated. We tested the Human3.6M dataset starting with the pre-trained *Stacked Hourglass* (SH) network (SH) to extract 2D point locations within the ground-truth bounding box, the results of which were further fine-tuned through the SH model [30]. Several automated methods without ground-truth bounding box were also investigated, including *ResNet-101-FPN* [23] with *Mask R-CNN* [16] and *Cascaded Pyramid Network* (CPN) [11]. Table 4 demonstrates the results with 2D directors by pre-trained SH, fine-tuned SH, and fine-tuned CPN models [35]. Further evaluation on 2D detectors can also be found in the second part of Table 1, where a comparison is shown with

| | Walk | | | Jog | | | |
|---|---|---|---|---|---|---|---|
| | S1 | S2 | S3 | S1 | S2 | S3 | Avg |
| Pavlakos et al. [34] | 22.3 | 19.5 | 29.7 | 28.9 | 21.9 | 23.8 | 24.4 |
| Martinez et al. [27]$*$ | 19.7 | 17.4 | 46.8 | 26.9 | 18.2 | 18.6 | 24.6 |
| Lee et al. [21] | 18.6 | 19.9 | 30.5 | 25.7 | 16.8 | 17.7 | 21.5 |
| Pavllo et al. [35] | 13.4 | 10.2 | 27.2 | 17.1 | 13.1 | 13.8 | 15.8 |
| Ours (n=27 CPN) | **13.1** | **9.8** | **26.8** | **16.9** | **12.8** | **13.3** | **15.4** |

Table 3: $Protocol\#2$ with P-MPJPE (mm): Reconstruction error on HumanEva. $(*)$ - single action model.

either the CPN estimation or the ground-truth (GT) as the input. For both cases, our attention model demonstrates clear advantages.

| Method | SH PT | SH FT | CPN FT | GT |
|---|---|---|---|---|
| Martinez et al. [27] | 67.5 | 62.9 | - | 45.5 |
| Hossain et al. [19] | - | 58.3 | - | 41.6 |
| Pavllo et al. [35] | 58.5 | 53.4 | 46.8 | 37.8 |
| ours(n=243) | 57.3 | 52.0 | 45.1 | 34.7 |
| Pavllo et al.[35] | - | - | 49.0 | - |
| Ours(n=27) | 62.5 | 56.4 | 49.4 | 39.7 |
| Ours(n=81) | 60.3 | 55.7 | 47.5 | 37.1 |
| Ours(n=243) | 59.2 | 54.9 | 46.7 | 35.5 |

Table 4: Top-table: Performance impacted by 2D detectors under $Protocol\#1$ with MPJPE (mm). Bottom-table: Causal sequence processing performance in terms of the different 2D detectors. PT - pre-trained, FT - fine-tuned, GT - ground-truth, SH - stacked hourglass, CPN - cascaded pyramid network.

**Causal Performance:** To facilitate real-time applications, we investigated the causal setting that has the architecture similar to the one described in Figure 2, but only considers the frames in the past. In the same manner, we implemented three prototypes with different receptive fields: $n = 27$, $n = 81$, and $n = 243$. Table 4 (bottom) demonstrates our causal model can still reach the same level of accuracy as state-of-the-art. For example, compared to the semi-supervised approach, the prototypes $n = 81$ and $n = 243$ yield smaller MPJPE [35]. It is worth mentioning even without the input of frames in the future, the temporal coherence is not compromised in the casual setting. The qualitative results are provided in our supplementary videos.

### 5.3. Ablation Studies

To verify the impact and performance of each component in the network, we conducted ablation experiments on the Human3.6M dataset under $Protocol\#1$.

**TCN Unit Channels:** we first investigated how the channel number $C$ affects the performance between TCN units and temporal attention models. In our test, we used both the CPN and GT as the 2D input. Starting with a receptive field of $n = 3 \times 3 \times 3 = 27$, as we increase the channels ($C \leq 512$), the MPJPE drops down significantly. However, the MPJPE changes slowly when $C$ grows between 512 and 1024, and remains almost stable afterwards. As shown in Figure 5, with the CPN input, a marginal improvement is yielded from MPJPE $49.9mm$ at $C = 1024$ to $49.6mm$ at $C = 2048$. A similar curve shape can be observed for the GT input. Considering the computation load with more parameters introduced, we chose $C = 1024$ in our experiments.

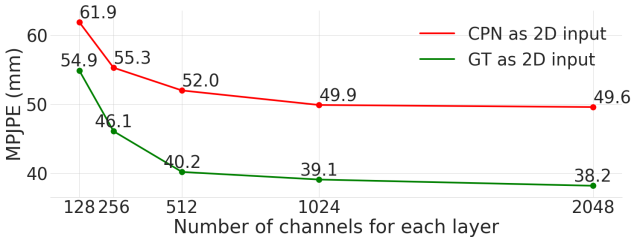



Figure 5: The impact of channel number on MPJPE. CPN: cascaded pyramid network and GT: ground-truth.

**Kernel Attention:** Table 5 shows how the setting of different parameters inside the Kernel Attention module impact the performance under $Protocol\#1$. The left three columns list the main variables. For validation purposes, we divide the configuration into three groups in row-wise. Within each group, we assign different values in one variable while keeping the other two fixed. The items in bold represent the best individual setting for each group. Empirically, we chose the combination of $M = 3$, $G = 8$, and $r = 128$ as the optimal setting (labeled in box). Note, we select $G = 8$ instead of the individual best assignment $G = 2$, which introduces a larger number of parameters with negligible MPJPE improvement.

| Kernels | Groups | Channels | Parameters | P1 |
|---|---|---|---|---|
| M=1 | G=1 | - | 16.95M | 37.8 |
| M=2 | G=8 | r=128 | 9.14M | 37.1 |
| **M=3** | G=8 | r=128 | 11.25M | **35.5** |
| M=4 | G=8 | r=128 | 13.36M | 38.0 |
| M=3 | G=1 | r=128 | 44.28M | 37.4 |
| M=3 | **G=2** | r=128 | 25.41M | **35.3** |
| M=3 | G=4 | r=128 | 15.97M | 35.6 |
| M=3 | $G = 8$ | r=128 | 11.25M | 35.5 |
| M=3 | G=16 | r=128 | 8.89M | 37.3 |
| M=3 | G=8 | r=64 | 10.20M | 35.9 |
| M=3 | G=8 | **r=128** | 11.25M | **35.5** |
| M=3 | G=8 | r=256 | 13.35M | 36.2 |

Table 5: Ablation study on different parameters in our kernel attention model. Here, we are using receptive field $n = 3 \times 3 \times 3 \times 3 \times 3 = 243$. The evaluation is performed on Human3.6M under $Protocol\#1$ with MPJPE (mm).

In Table 6, we discuss the choice of different types of receptive fields and how it affects the network performance. The first column shows various layer configurations, which generates different receptive fields, ranging from $n = 27$ to $n = 1029$. To validate the impact of $n$, we fix the other parameters, i.e. $M = 3$, $G = 8$, $r = 128$. Note that for

a network with lower number of layers (e.g. $L = 3$), a larger receptive field may reduce the error more effectively. For example, increasing the receptive field from $n = 3 \times 3 \times 3 = 27$ to $n = 3 \times 3 \times 7 = 147$, the MPJPE drops from 40.6 to 36.8 . However, for a deeper network, a larger receptive field may not be always optimal, e.g. when $n = 1029$, MPJPE $= 37.0$. Empirically, we obtained the best performance with the setting of $n = 243$ and $L = 5$, as indicated in the last row.

| Receptive fields | Kernels | Groups | Channels | Parameters | P1 |
|---|---|---|---|---|---|
| $3\times3\times3=27$ | M=1 | G=1 | - | 8.56M | 40.6 |
| $3\times3\times3=27$ | M=2 | G=4 | r=128 | 6.21M | 40.0 |
| $3\times5\times3=45$ | M=2 | G=4 | r=128 | 6.21M | 39.9 |
| $3\times5\times5=75$ | M=2 | G=4 | r=128 | 6.21M | 38.5 |
| $3\times3\times3=27$ | M=3 | G=8 | r=128 | 5.69M | 39.5 |
| $3\times5\times3=45$ | M=3 | G=8 | r=128 | 5.69M | 39.2 |
| $3\times5\times5=75$ | M=3 | G=8 | r=128 | 5.69M | 38.2 |
| $3\times7\times7=147$ | M=3 | G=8 | r=128 | 5.69M | 36.8 |
| $3\times3\times3\times3=81$ | M=3 | G=8 | r=128 | 8.46M | 37.8 |
| $3\times5\times5\times5=375$ | M=3 | G=8 | r =128 | 8.46M | 36.6 |
| $3\times7\times7\times7=1029$ | M=3 | G=8 | r=128 | 8.46M | 37.0 |
| **$3\times3\times3\times3\times3=243$** | M=3 | G=8 | r=128 | 11.25M | **35.5** |

Table 6: Ablation study on different receptive fields in our kernel attention model. The evaluation is performed on Human3.6M under $Protocol\#1$ with MPJPE (mm).

**Multi-Scale Dilation:** To evaluate the impact of the dilation component on the network, we tested the system with and without dilation and compared their individual outcomes. In the same way, the GT and CPN 2D detectors are used as input and being tested on the Human3.6M dataset under $Protocol\#1$. Table 7 demonstrates the integration of attention, and multi-scale dilation components surpass their individual performance with the minimum MPJPE for all the three prototypes. We also found the attention model makes an increasingly significant contribution as the layer number grows. This is because more layers lead to a larger receptive field, allowing the multi-scale dilation to capture long-term dependency across frames. The effect is more noticeable when fast motion or self-occlusion present in videos.

**Qualitative Results** We also further evaluate our approach on a number of challenging wide videos, such as activities of fast motion or low-resolution human images, which are extremely difficult to obtain a meaningful 2D detection. For example, in Figure 6, the person playing sword not only has quick body movement also has a long casual dress with partial occlusion; the skating girl has fast speed generating blur regions. Our approach achieves a high level of robustness and accuracy in these challenging scenarios. More results can be found in the supplementary material.

| Method \ Model | $n = 27$ | $n = 81$ | $n = 243$ |
|---|---|---|---|
| Attention model (CPN) | 49.1 | 47.2 | 46.3 |
| Multi-Scale Dilation model (CPN) | 48.7 | 47.1 | 45.7 |
| Attention and Dilation (CPN) | 48.5 | 46.3 | 45.1 |
| Attention model (GT) | 39.5 | 37.8 | 35.5 |
| Multi-Scale Dilation model (GT) | 39.3 | 37.2 | 35.3 |
| Attention and Dilation (GT) | 38.9 | 36.2 | 34.7 |

Table 7: Ablation study on different components in our method. The evaluation is performed on Human3.6M under $Protocol\#1$ with MPJPE (mm).

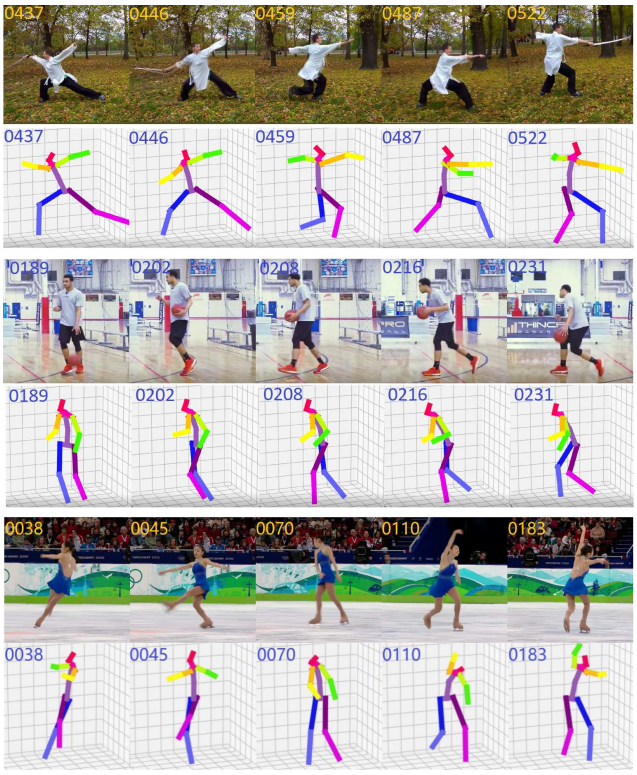



Figure 6: Qualitative results on wide videos.

## 6. Conclusion

We presented an attentional approach for 3D pose estimation from 2D videos. Combining multi-scale dilation with the temporal attention module, our system is able to capture long-range temporal relationships across frames, thereby significantly enhancing temporal coherency. Our experiments show a robust, high-fidelity prediction that compares favorably to related techniques. We believe our system substantially advances the state-of-the-art in video-based 3D pose estimation, making it practical for real-time applications.

## References


[1] S. Amin, M. Andriluka, M. Rohrbach, and B. Schiele. Multiview pictorial structures for 3d human pose estimation. *BMVC*, 2013.

[2] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2009.

[3] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2016.

[4] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[5] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. *European Conference on Computer Vision (ECCV)*, page 1–18, 2016.

[6] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[7] V. Olaru C. Ionescu, D. Papava and C. Sminchisescu. 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.

[8] C.-H. Chen and D. Ramanan. 3d human pose estimation = 2d pose estimation + matching. *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7035–7043, 2017.

[9] W. Chen, H. Wang, Y. Li, and H. Su et al. Synthesizing training images for boosting human 3d pose estimation. *Fourth International Conference on 3D Vision (3DV)*, pages 479–488, 2016.

[10] Y. Chen, C. Shen, H. Chen, X. S. Wei, L. Liu, and J. Yang. Adversarial learning of structure-aware fully convolutional networks for landmark localization. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[11] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7103–7112, 2018.

[12] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio. Attention-based models for speech recognition. *Advances in Neural Information Processing Systems 28*, pages 577–585, 2015.

[13] Rishabh Dabral, Anurag Mundhada, Uday Kusupati, Safeer Afaque, Abhishek Sharma, and Arjun Jain. Learning 3d human pose from structure and motion. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 668–683, 2018.

[14] Hao-Shu Fang, Yuanlu Xu, Wenguan Wang, Xiaobai Liu, and Song-Chun Zhu. Learning pose grammar to encode human body configuration for 3d pose estimation. *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[15] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Pose search: Retrieving people using their pose. *IEEE Conference on Computer Vision and Pattern Recognition*, page 1–8, 2009.

[16] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask r-cnn. *International Conference on Computer Vision (ICCV)*, page 2980–2988, 2017.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[18] S. Hochreiter and J. Schmidhuber. Long short-term memory. neural computation. *Neural computation*, 9(8):1735 –1780, 1997.

[19] M. Hossain and J. Little. Exploiting temporal information for 3d human pose estimation. *European Conference on Computer Vision (ECCV)*, pages 69–86, 2018.

[20] I. Kostrikov and J. Gall. Depth sweep regression forests for estimating 3d human pose from images. *British Machine Vision Conference (BMVC)*, 2014.

[21] Kyoungoh Lee, Inwoong Lee, and Sanghoon Lee. Propagating lstm: 3d pose estimation based on joint interdependency. *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–135, 2018.

[22] S. Li, W. Zhang, and A. B. Chan. Maximum-margin structured learning with deep networks for 3d human pose estimation. *International Conference on Computer Vision (ICCV)*, page 2848–2856, 2015.

[23] T. Lin, P. Dollar, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. *Conference on Computer Vision and Pattern Recognition (CVPR)*, page 936–944, 2017.

[24] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*, 2019.

[25] Diogo C Luvizon, David Picard, and Hedi Tabia. 2d/3d pose estimation and action recognition using multitask deep learning. pages 5137–5146, 2018.

[26] C. Mandery, O. Terlemez, M. Do, N. Vahrenkamp, and T. Asfour. The kit whole-body human motion database. *International Conference on Advanced Robotics (ICAR)*, pages 329–336, 2015.

[27] J. Martinez, R. Hossain, J. Romero, and J. J. Little. A simple yet effective baseline for 3d human pose estimation. *International Conference on Computer Vision (ICCV)*, page 2659–2668, 2017.

[28] Diganta Misra. Mish: A self regularized non-monotonic neural activation function. *arXiv preprint arXiv:1908.08681*, 2019.

[29] N. Neverova, C. Wolf, G. W. Taylor, and F. Nebout. Multi-scale deep learning for gesture detection and localization. *European Conference on Computer Vision (ECCV) Workshops*, pages 474–490, 2014.

[30] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. *European Conference on Computer Vision*, pages 483–499, 2016.

[31] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner,

Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[32] C. Palmero, A. Clapés, C. Bahnsen, A. Møgelmose, T. B. Moeslund, and S. Escalera. Multi-modal rgb–depth–thermal human body segmentation. *International Journal of Computer Vision*, 118(2):217–239, 2016.

[33] S. Park, J. Hwang, and N. Kwak. 3d human pose estimation using convolutional neural networks with 2d pose information. *European Conference on Computer Vision (ECCV) Workshops*, page 156–169, 2016.

[34] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. *Conference on Computer Vision and Pattern Recognition (CVPR)*, page 1263–1272, 2017.

[35] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7753–7762, 2019.

[36] Tim Rocktäsche, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. Reasoning about entailment with neural attention. In *ICLR*, 2016.

[37] D. Ruck, S. Rogers, and M. Kabrisky. Feature selection using a multilayer perceptron. *Journal of Neural Network Computing*, 2(2):40–48, 1990.

[38] N. Sarafianos, B. Boteanu, B. Ionescu, and I. A. Kakadiaris. 3d human pose estimation: A review of the literature and analysis of covariates. *CVIU*, page 1–20, 2016.

[39] L. Sigal, A. O. Balan, and M. J. Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision*, 87(12):4–27, 2010.

[40] B. Tekin, A. Rozantsev, V. Lepetit, and P. Fua. Direct prediction of 3d body poses from motion compensated sequences. *Conference on Computer Vision and Pattern Recognition (CVPR)*, page 991–1000, 2016.

[41] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1653–1660, 2014.

[42] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid. Learning from synthetic humans. *Conference on Computer Vision and Pattern Recognition (CVPR)*, page 109–117, 2017.

[43] W. Yang, W. Ouyang, X. Wang, J. Ren, H. Li, and X. Wang. 3d human pose estimation in the wild by adversarial learning. *Conference on Computer Vision and Pattern Recognition (CVPR)*, page 5255–5264, 2018.

[44] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. *Conference on Computer Vision and Pattern Recognition (CVPR)*, page 1385–1392, 2011.

[45] W. Yin, H. Schütze, B. Xiang, and B. Zhou. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *Transactions of the Association for Computational Linguistics*, 4:259–272, 2016.

[46] J. Yoo and T. Han. Fast normalized cross-correlation. *Circuits, Systems and Signal Processing*, 28(819):1–13, 2009.

[47] Michael R Zhang, James Lucas, Geoffrey Hinton, and Jimmy Ba. Lookahead optimizer: k steps forward, 1 step back. *arXiv preprint arXiv:1907.08610*, 2019.

[48] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3425–3435, 2019.

[49] X. Zhou, X. Sun, W. Zhang, S. Liang, and Y. Wei. Deep kinematic pose regression. *European Conference on Computer Vision (ECCV) Workshops*, page 156–169, 2016.

[50] X. Zhou, M. Zhu, S. Leonardos, K. G. Derpanis, and K. Daniilidis. Sparseness meets deepness: 3d human pose estimation from monocular video. *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4966–4975, 2016.