2D to 3D Pose Lifting

Amirhosein Ghasemabadi
University of Alberta
ghasemab@ualberta.ca

Mohammadali Shakerdargah
University of Alberta
shakerda@ualberta.ca

Mohammadamin Samadi Khoshkhoo
University of Alberta
msamadik@ualberta.ca

1. Abstract

The task of 3D human pose estimation involves estimating the 3D joint locations of a human body from 2D images or videos. This task has received significant attention in recent years due to its role in various applications, such as clinical applications, computer animation, action recognition, and human-robot interaction. Many state-of-the-art approaches use a two-stage pipeline, in which 2D key points are first estimated and then lifted to 3D space. However, this 2D-to-3D lifting method is still a difficult problem because of the inherent ambiguity in depth. To address this difficulty, researchers have explored using temporal context information to improve performance. One of the most successful models in this area is the Vanilla Transformer, which is capable of capturing long-range dependencies. However, this model has limitations, including a small temporal receptive field and limited temporal correlation windows. To overcome these limitations, the Strided Transformer Encoder (STE) is introduced, which is a modified Transformer that gradually merges nearby poses to shrink the sequence length and uses strided convolutions to reduce the sequence length. The STE is able to model both global and local information in a hierarchical architecture, allowing for improved performance on 3D human pose estimation tasks.Moreover, with a change in the attention layer, we made the model even faster.

2. Introduction

The task of 3D human pose estimation involves estimating the 3D joint locations of a human body from 2D images or videos.[1]-[4] This task has received significant attention in recent years due to its role in a variety of applications, such as clinical applications [5], computer animation [6], action recognition [7], and human-robot interaction [8]. Many state-of-the-art approaches use a two-stage pipeline, in which 2D keypoints are first estimated and then lifted to 3D space. While this 2D-to-3D lifting method [9] benefits from the performance of 2D pose detectors, it is still a difficult problem because of the inherent ambiguity in depth, as multiple 3D interpretations can be projected to the same 2D pose in image space.

To address the difficulty of the 2D-to-3D lifting method, many researchers have explored using temporal context information. [10] Some methods use past and future data in a sequence to predict the 3D pose of the target frame. For example, Cai et al.[11] proposed a local-to-global graph convolutional network to exploit spatial and temporal relations to estimate 3D keypoints from a 2D pose sequence. However, these approaches have small temporal receptive fields and limited temporal correlation windows, which makes it difficult to model long-range dependencies.

There are newer and more complicated networks that acquire attention mechanism and show better performance for these types of tasks. One of the most relatable neural networks is Vanilla Transformer [12] that is a model which is designed to capture long-range dependencies [13] and has been successful in huge computer vision tasks[14]. It is composed of a self-attention module and a position-wise feed-forward network (FFN). An attention function maps a query and a set of key-value pairs to an output vector. The output is calculated as a weighted sum of the values, where the weight assigned to each value is determined by a compatibility function that compares the query to the corresponding key. Both the query, keys, values, and output are vectors. Instead of using a single attention function with keys that have the same dimension as dimension of the model, values, and queries, it is more beneficial to linearly project the queries, keys, and values h times with different, learned linear projections. We then perform the attention function in parallel on each of these projected versions of queries, keys, and values, yielding of the proposed dimension output values. These are concatenated and then projected again, resulting in the final values.

The Vanilla Transformer Encoder (VTE) [12] appears to be a viable choice for 2D-to-3D pose lifting in order to capture long-range dependencies, but it has several limitations. These include (i) the full-length sequence in the forward pass containing too much redundancy; (ii) the time and memory complexity of the attention operation growing quadratically with the input length, resulting in a small receptive field; and [16] (iii) being less capable of extracting fine-grained local feature patterns. To address these issues, the Strided Transformer Encoder (STE) is introduced which is a modified Transformer which gradually merges nearby poses to shrink the sequence length and replaces the fully-connected layers in FFN with strided convolutions to progressively reduce the sequence length [13]. With the proposed STE, we can model both global and local information in a hierarchical architecture, and the computation in FFN can be traded off for constructing a deeper model to boost the model capacity.

Despite the ability of the STE to aggregate long-range information into a single-pose representation, it remains to be seen whether this single representation is sufficient to represent a long sequence and how it can be used to improve performance. We have observed that supervising the model at only a single target frame scale tends to break temporal smoothness among frames, while supervising only at a full sequence scale cannot explicitly learn a specific representation for the target frame. This has led us to develop a method that can effectively embed both scales into a learnable framework. Thus, based on the outputs of VTE and STE, we propose a full-to-single supervision scheme at both full and single scales, which can impose additional temporal smoothness constraints at the full sequence scale and refine the estimation at the single target frame scale. This scheme yields smoother and more accurate 3D poses. The proposed architecture is called Strided Transformer, as depicted in Fig. 3. We conducted extensive experiments on two standard 3D human pose estimation datasets, Human3.6M [17] and HumanEva-I [18], and the results show that it achieves near state-of-the-art performance. Our contributions can be summarized as follows:

- We acquired a new Transformer-based architecture for 3D human pose estimation called Strided Transformer and modified the attention mechanism to improve model's performance. It can effectively lift a long 2D pose sequence to a single 3D pose, reduce the sequence redundancy and computation cost,

- A full-to-single supervision scheme is designed to impose additional temporal smoothness constraints during training at the full sequence scale and further improve the estimation at the single target frame scale.

- To improve the performance of our model, we have modified the loss function to include weights for different frames. This allows the model to better capture the relative importance of different frames and to learn more effectively during training. By incorporating these weights into the loss function, we have been able to improve the performance of our model.

Our model and architecture have been designed to achieve near state-of-the-art performance. Through the implementation of advanced techniques and the careful selection of appropriate parameters, we are confident that our approach will yield results that are competitive with the current state of the art. By leveraging the full potential of our model and architecture, we believe that we can achieve a level of performance that performs faster than usual with competitive results.

3. Literature Review

Early approaches to using deep neural networks for 3D pose estimation often involved directly learning a mapping from RGB images to 3D poses in a single stage. However, these methods required complex architectures and were computationally expensive, making them impractical for real-world applications [19].

One proposed work is a unified formulation for the problem of 3D human pose estimation from a single raw RGB image.[3] This approach jointly reasons about 2D joint estimation and 3D pose reconstruction in order to improve both tasks. The method uses an integrated approach that combines probabilistic knowledge of 3D human pose with a multi-stage convolutional neural network (CNN) architecture [25]. This approach leverages knowledge of plausible 3D landmark locations to refine the search for better 2D sites, resulting in substantially better performance than other methods. On average, this approach shows a 4.7mm improvement over the next best competitor at the time.

One architecture called a Semantic Graph Convolutional Network (SGCN) that captures semantic information and applies it to the problem of regressing 3D human pose from a single image has a great reputation in the problems of pose lifting [28]. This method uses graph convolutional networks (GCNs) instead of traditional convolutional neural networks (CNNs) because GCNs are better suited to tasks with irregular input structures, such as the 3D human pose regression problem. The SGCN architecture is specifically designed for use with single image inputs, not videos. Another framework for improving the 3D pose distribution of an initial training set is Cascaded Deep Monocular 3D Human Pose Estimation [23]. A novel cascaded 3D human pose estimation model is trained using this framework, achieving state-of-the-art performance for single-frame 3D human pose estimation.

Another proposed architecture is SRNet which is a split-and-recombine approach that improves the generalization of 3D human pose estimation networks[26]. It is difficult for networks to predict human poses that are rare or unseen in the training set. SRNet addresses this challenge by using a Dynamical Graph Network (DG-Net) that can dynamically identify human-joint affinity and estimate 3D pose by adaptively learning spatial and temporal joint relations from videos. The Dynamical Spatial/Temporal Graph convolution used by SRNet discovers the spatial and temporal human-joint affinity for each video exemplar, depending on the spatial distance and temporal movement similarity between human joints in the video. This allows SRNet to effectively understand which joints are spatially closer and have a consistent motion, reducing depth ambiguity and movement uncertainty when lifting 2D poses to 3D poses.

Another architecture that was proposed recently is MixSTE which is a seq2seq mixed spatio-temporal encoder for 3D human pose estimation in video. Some recent papers consider the body joints in all frames globally to learn spatio-temporal correlations [27]. MixSTE proposes a temporal transformer block to model each joint's temporal motion and a spatial transformer block to learn inter-joint spatial correlations. These two blocks are used alternately to obtain better spatio-temporal feature encoding. The network output is also extended from the central frame to the entire input video, improving the coherence between the input and output sequences.

4. Proposed Method

In this section, we provide an overview of the proposed modified Strided Transformer for 3D human pose estimation from a 2D video stream. We then demonstrate how our Transformer-based architecture learns a representative single-pose representation from redundant sequences, resulting in improved estimation.

4.1. Overview

Fig. 3 shows the overall framework of our proposed method. Given a sequence of the estimated $2D$ poses $P = \{p_1, \ldots, p_T\}$ from videos, we aim at reconstructing 3D joint locations $X \in \mathbb{R}^{J\times 3}$ for a target frame (center frame), where $p_t \in \mathbb{R}^{J\times 2}$ denotes the 2D joint locations at frame t, T is the number of video frames, and J is the number of joints.

In this paper, we use the same code and the network of based paper [24] with a few minor modifications.

The proposed network includes a Vanilla Transformer Encoder (VTE) and a Modified Strided Transformer Encoder (STE). The VTE is used to model long-range information and is trained using a full sequence scale to enforce temporal smoothness. The STE then aggregates the information to generate a single target pose representation, which is trained using a single target frame scale to produce more accurate estimates. The network is trained using a full-to-single prediction scheme at both the full sequence and single target frame scales.

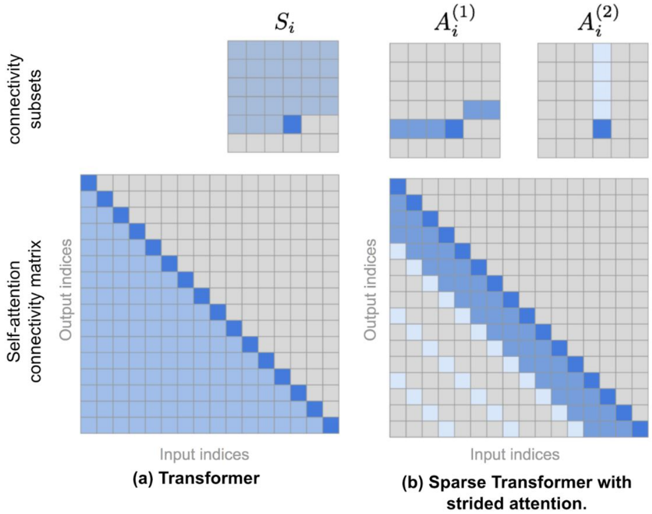


Figure 1. The top row illustrates the attention connectivity patterns in (a) Transformer, and (b) Sparse Transformer with strided attention. The bottom row contains corresponding self-attention connectivity matrices. Note that the top and bottom rows are not in the same scale. (Image source: [29] + a few of extra annotations.).

4.2. Modified Strided Transformer Encoder

Despite the success of Transformers [12] in many computer vision tasks, their use is limited in video-based tasks that require a single-vector representation of a sequence. Strided Transformer Encoder (STE) was introduced, which gradually compresses the sequence of hidden states and models global and local information in a hierarchical architecture. Each layer of the proposed STE consists of a multi-head self-attention (MSA) [12] module and a convolutional feed-forward network (CFFN). In this paper, we substituted the attention layer with strided attention to make the model simpler and faster.

1) Sparse Attention Matrix Factorization (Strided Attention)

The compute and memory cost of the vanilla Transformer grows quadratically with sequence length and thus it is hard to be applied on very long sequences.

Sparse Transformer [29] introduced factorized self-attention, through sparse matrix factorization, making it possible to train dense attention networks with hundreds of layers on sequence length up to 16,384, which would be infeasible on modern hardware otherwise.

A self-attention layer maps a matrix of input embeddings X to an output matrix and is parameterized by a connectivity pattern $S = \{S_1, \ldots, S_n\}$, where S_i denotes the set of indices of the input vectors to which the i th output vector attends. The output vector is a weighted sum of transformations of the input vectors:

$$\operatorname{Attend}(X, S) = \left(a\left(\mathbf{x}_i, S_i\right)\right)_{i \in \{1, \ldots, n\}} \tag{1}$$

$$a\left(\mathbf{x}_i, S_i\right) = \operatorname{softmax}\left(\frac{\left(W_q \mathbf{x}_i\right) K_{S_i}^T}{\sqrt{d}}\right) V_{S_i} \tag{2}$$

$$K_{S_i} = \left(W_k \mathbf{x}_j\right)_{j \in S_i} \quad V_{S_i} = \left(W_v \mathbf{x}_j\right)_{j \in S_i} \tag{3}$$

Full self-attention for autoregressive models defines $S_i = \{j : j \leq i\}$, allowing every element to attend to all previous positions and its own position.

Factorized self-attention instead has p separate attention heads, where the m th head defines a subset of the indices $A_i^{(m)} \subset \{j : j \leq i\}$ and lets $S_i = A_i^{(m)}$. We are chiefly interested in efficient choices for the subset A, where $\left|A_i^{(m)}\right| \propto \sqrt[p]{n}$. Strided attention is one of the two proposed types of factorized attention.

A natural approach to defining a factorized attention pattern in two dimensions is to have one head attend to the previous l locations and the other head attend to every l th location, where l is the stride and chosen to be close to $\sqrt{n}$, a method we call strided attention.

$$A_i^{(1)} = \{t, t+1, \ldots, i\} \text{ for } t = \max(0, i-l) \tag{4}$$

$$A_i^{(2)} = \{j : (i-j) \bmod l = 0\} \tag{5}$$

It is easier to understand the concepts as illustrated in Fig. 1 with 2D image inputs as examples.

2) Convolutional feed-forward network In the existing fully-connected (FC) layers in the FFN of VTE, a full-length sequence of hidden representations is maintained across all layers, resulting in high computation costs. This is redundant for video-based pose estimation, as nearby poses are similar. To more accurately reconstruct the 3D body joints of the target frame, important information should be selectively aggregated from the entire pose sequence.

To tackle this issue, inspired by the previous works [9], [16] that employ temporal convolutions to shrink the sequence length effectively, An modification to the generic FFN was suggested by the base paper. Given the input feature vector $Z \in \mathbb{R}^{T \times D_{in}}$ with T sequences and D_{in} channels to generate an output of $\left(\tilde{T}, D_{\text{out}}\right)$ features, the operation performed by FC in FFN can be formulated as:

$$\mathrm{FC}_{t, d_{\text{out}}}(z) = \sum_{i}^{D_{\text{in}}} w_{d_{\text{out}}, i} * z_{t, i}. \tag{6}$$

If 1D convolution is considered with kernel size K and strided factor S, a strided convolution in CFFN can be computed as:

$$\operatorname{Conv}_{S(t), c_{\text{out}}}(z) = \sum_{i}^{D_{\text{in}}} \sum_{k}^{K} w_{d_{\text{out}}, i, k} * z_{S\left(t - \frac{K-1}{2} + k\right), i}. \tag{7}$$

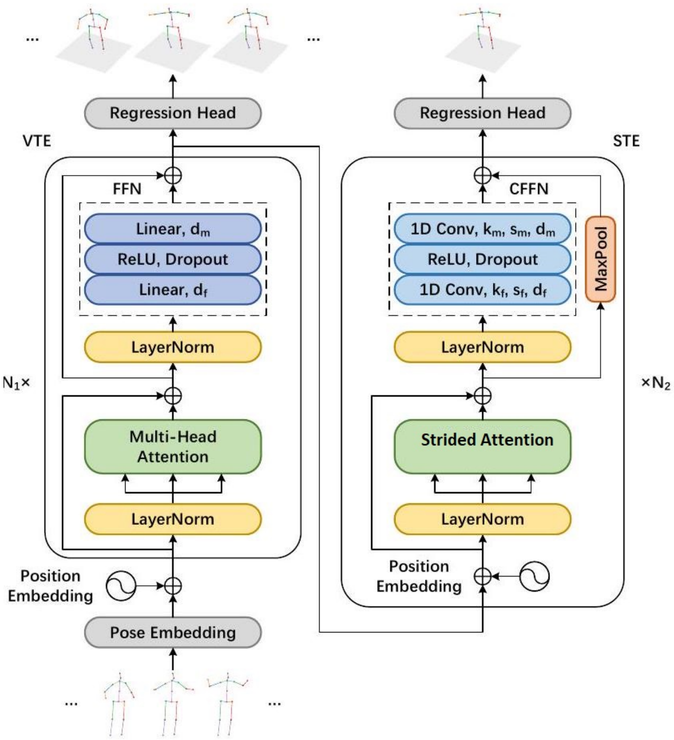


Figure 2. The network architecture of our proposed Strided Transformer. The left is the VTE and the right is the STE. Here, N_1 and N_2 denote the number of layers of the two modules, respectively. The hyperparameters k, s, d_m and d_f are the kernel size, the strided factor, the dimension, and the number of hidden units. The max pooling operation is applied to the residuals to match the temporal dimensions.(Image source: [24] + slight modification)

In this way, fully-connected layers in FFN of VTE are replaced with strided convolutions. The modified VTE is termed as Strided Transformer Encoder (STE), which can be represented as:

$$\hat{Z}^{n-1} = Z^{n-1} + \operatorname{MSA}\left(\operatorname{LN}\left(Z^{n-1}\right)\right) \tag{8}$$

$$Z^n = \operatorname{MaxPool}\left(\hat{Z}^{n-1}\right) + \operatorname{CFFN}\left(\operatorname{LN}\left(\hat{Z}^{n-1}\right)\right), \tag{9}$$

where $\operatorname{LN}(\cdot)$ denotes the layer normalization, $\operatorname{MaxPool}(\cdot)$ denotes the max pooling operation, and $n \in [1, \ldots, N]$ is the index of STE layers.

As shown in Fig. 2, the STE has a hierarchical global and local architecture. The self-attention mechanism models global context, while strided convolution captures local context. As illustrated in Fig. 3, this architecture gradually merges nearby poses into a short sequence length representation. Through this hierarchical design, both redundancies in the sequence and computation costs can be reduced.

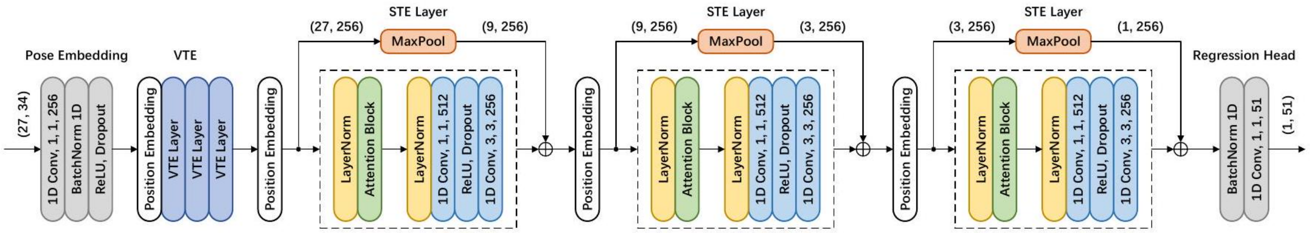


Figure 3. An instantiation of the proposed Strided Transformer network. It reconstructs the target 3D body joints by progressively reducing the sequence length. The input consists of 2D keypoints for a receptive field of 27 frames with $J = 17$ joints. Convolutional feed-forward networks are in blue where $(3, 3, 256)$ denotes kernels of size 3 with strided factor 3 and 256 output channels. The tensor sizes are shown in parentheses, e.g., $(27, 34)$ denotes 27 frames and 34 channels. Due to strided convolutions, the max pooling operation is applied to the residuals to match the shape of subsequent tensors.(Image source: [24])

4.3. Network Architecture

The overall network archtecture is the same the base paper[24] with a change in the attention unit. As shown in Fig. 3, the proposed network is composed of four components: a pose embedding, a Vanilla Transformer Encoder (VTE), a Strided Transformer Encoder (STE), and a regression head.

1) Pose embedding: Given a sequence of the estimated 2D poses $P \in \mathbb{R}^{T \times J \times 2}$, the pose embedding first concatenates (x, y) coordinates of the J joints for each frame to tokens $P' \in \mathbb{R}^{T \times (J \cdot 2)}$, and then embeds each token to a high dimensional feature $Z_0 \in \mathbb{R}^{T \times d_m}$ using a 1D convolutional layer with d_m channels, followed by batch normalization, dropout, and a ReLU activation.

2) Vanilla Transformer Encoder: Suppose that the VTE consists of N_1 layers, the learnable position embedding $E_1 \in \mathbb{R}^{T \times d_m}$ is used before the first layer of VTE, which can be formulated as follows:

$$Z_1^0 = Z_0 + E_1. \tag{10}$$

Then, given the embedded feature Z_1^0, the VTE layers can be represented as:

$$\hat{Z}_1^{n-1} = Z_1^{n-1} + \mathrm{MSA}\left(\mathrm{LN}\left(Z_1^{n-1}\right)\right), \tag{11}$$

$$Z_1^n = \hat{Z}_1^{n-1} + \mathrm{FFN}\left(\mathrm{LN}\left(\hat{Z}_1^{n-1}\right)\right), \tag{12}$$

where $n \in [1, \ldots, N_1]$ is the index of VTE layers. It can be expressed by using a function of a VTE layer $\mathrm{VTE}(\cdot)$:

$$Z_1^n = \mathrm{VTE}\left(Z_1^{n-1}\right) \tag{13}$$

3) Strided Transformer Encoder: For the STE, it is built upon the outputs of VTE and takes the $Z_1^{N_1} \in \mathbb{R}^{T \times d_m}$ as input. The learnable position embeddings $E_2 \in \mathbb{R}^{S(t) \times d_m}$ with strided factor S are used for every layer of STE due to the different sequence lengths. Then, the STE layers can be represented as follows:

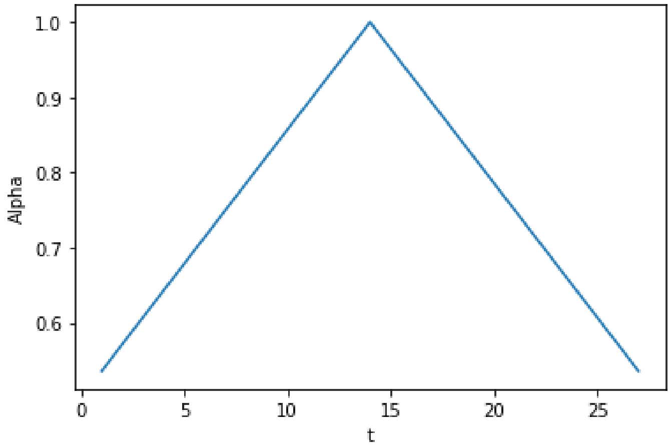


Figure 4. α_t values for the loss function with the middle frame as the target.

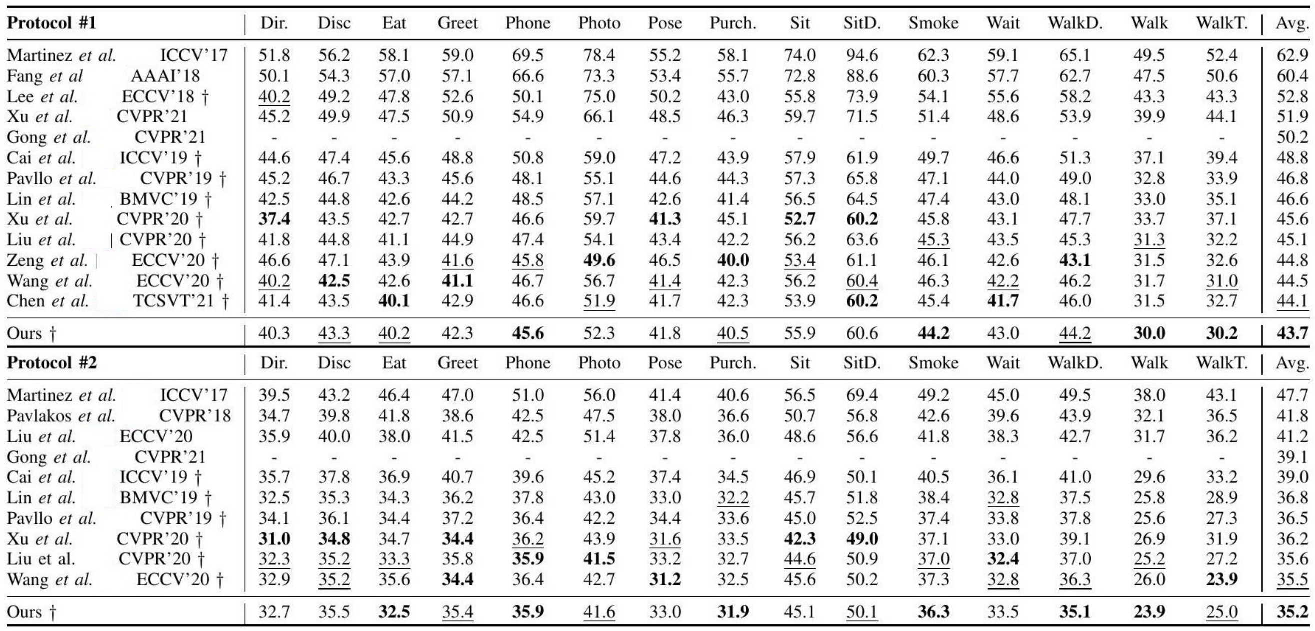
$$Z_2^n = \mathrm{STE}\left(Z_2^{n-1} + E_2^n\right), \tag{14}$$

where $n \in [1, \ldots, N_2]$ is the index of STE layers, $Z_2^0 = Z_1^{N_1}$, and $\mathrm{STE}(\cdot)$ denotes the function of an STE layer whose details can be found in Eq. (8) and Eq. (9).

4) Regression head: In order to perform the regression, a batch normalization and a 1D convolutional layer are applied to the outputs of VTE and STE, $Z_1^{N_1} \in \mathbb{R}^{T \times d_m}$ and $Z_2^{N_2} \in \mathbb{R}^{1 \times d_m}$, respectively. Finally, the outputs of 3D pose prediction are $\tilde{X}$ and X, where $\tilde{X} \in \mathbb{R}^{T \times J \times 3}$ and $X \in \mathbb{R}^{J \times 3}$ are predictions of the 3D pose sequence and the 3D joint locations of the target frame, respectively.

4.4. Full-to-Single Prediction

The base paper proposed a full-to-single scheme to incorporate both full sequence and single target frame scale constraints into the framework [11], [19]. This scheme further refines the intermediate predictions to produce more accurate estimations rather than using a single component with a single output. More precisely, the full sequence



Protocol #1		Dir.	Disc	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Martinez *et al.*	ICCV'17	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	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
Lee *et al.*	ECCV'18 †	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
Xu *et al.*	CVPR'21	45.2	49.9	47.5	50.9	54.9	66.1	48.5	46.3	59.7	71.5	51.4	48.6	53.9	39.9	44.1	51.9
Gong *et al.*	CVPR'21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	50.2
Cai *et al.*	ICCV'19 †	44.6	47.4	45.6	48.8	50.8	59.0	47.2	43.9	57.9	61.9	49.7	46.6	51.3	37.1	39.4	48.8
Pavllo *et al.*	CVPR'19 †	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
Lin *et al.*	BMVC'19 †	42.5	44.8	42.6	44.2	48.5	57.1	42.6	41.4	56.5	64.5	47.4	43.0	48.1	33.0	35.1	46.6
Xu *et al.*	CVPR'20 †	**37.4**	43.5	42.7	42.7	46.6	59.7	**41.3**	45.1	**52.7**	**60.2**	45.8	43.1	47.7	33.7	37.1	45.6
Liu *et al.*	CVPR'20 †	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
Zeng *et al.*	ECCV'20 †	46.6	47.1	43.9	41.6	45.8	**49.6**	46.5	**40.0**	53.4	61.1	46.1	42.6	**43.1**	31.5	32.6	44.8
Wang *et al.*	ECCV'20 †	40.2	**42.5**	42.6	**41.1**	46.7	56.7	41.4	42.3	56.2	60.4	46.3	42.2	46.2	31.7	31.0	44.5
Chen *et al.*	TCSVT'21 †	41.4	43.5	**40.1**	42.9	46.6	51.9	41.7	42.3	53.9	**60.2**	45.4	**41.7**	46.0	31.5	32.7	44.1
Ours †		40.3	43.3	40.2	42.3	**45.6**	52.3	41.8	40.5	55.9	60.6	**44.2**	43.0	44.2	**30.0**	**30.2**	**43.7**
Protocol #2		Dir.	Disc	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Martinez *et al.*	ICCV'17	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
Pavlakos *et al.*	CVPR'18	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
Liu *et al.*	ECCV'20	35.9	40.0	38.0	41.5	42.5	51.4	37.8	36.0	48.6	56.6	41.8	38.3	42.7	31.7	36.2	41.2
Gong *et al.*	CVPR'21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	39.1
Cai *et al.*	ICCV'19 †	35.7	37.8	36.9	40.7	39.6	45.2	37.4	34.5	46.9	50.1	40.5	36.1	41.0	29.6	33.2	39.0
Lin *et al.*	BMVC'19 †	32.5	35.3	34.3	36.2	37.8	43.0	33.0	32.2	45.7	51.8	38.4	32.8	37.5	25.8	28.9	36.8
Pavllo *et al.*	CVPR'19 †	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
Xu *et al.*	CVPR'20 †	**31.0**	**34.8**	34.7	**34.4**	36.2	43.9	31.6	33.5	**42.3**	**49.0**	37.1	33.0	39.1	26.9	31.9	36.2
Liu et al.	CVPR'20 †	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
Wang *et al.*	ECCV'20 †	32.9	35.2	35.6	**34.4**	36.4	42.7	**31.2**	32.5	45.6	50.2	37.3	32.8	36.3	26.0	**23.9**	35.5
Ours †		32.7	35.5	**32.5**	35.4	**35.9**	41.6	33.0	**31.9**	45.1	50.1	**36.3**	33.5	**35.1**	**23.9**	25.0	**35.2**

Figure 5. Quantitative comparisons with state-of-the-art methods in different receptive fields on Human3.6M. The computational complexity, MPJPE, and frame per second (FPS) are reported. FPS is computed on a single GeForce GTX 2080 Ti GPU.(Image source: [24])

scale can enforce temporal smoothness, and the single target frame scale helps learn a specific representation for the target frame.

1) Full sequence scale: The first step is to supervise at full sequence scale by imposing extra temporal smoothness constraints during training from the output of VTE followed by a regression head. A sequence loss $\mathcal{L}_f$ is used to improve upon single frame predictions for temporal consistency over a sequence. This loss ensures that the estimated 3D pose sequences $\tilde{X} \in \mathbb{R}^{T\times J\times 3}$ coincide with the ground truth 3D joint sequences $Y \in \mathbb{R}^{T\times J\times 3}$:

$$\mathcal{L}_f = \sum_{t=1}^{T}\sum_{i=1}^{J} \alpha_t \left\|Y_i^t - \tilde{X}_i^t\right\|_2, \tag{15}$$

where $\tilde{X}_i^t$ and Y_i^t represent the sequence of estimated 3D poses and ground truth 3D joint locations of joint i at frame t, respectively.
α_t was added to the loss function to force the model to pay more attention to the target frame and its closest neighbor frames.

2) Single target frame scale: In the second step, the supervision is adopted on the output of STE followed by a regression head. A single-frame loss $\mathcal{L}_s$ is used to refine the estimation at the single target frame scale. It minimizes the distance between the estimated 3D pose $X \in \mathbb{R}^{J\times 3}$ and the target ground truth 3D joint annotation $Y \in \mathbb{R}^{J\times 3}$:

$$\mathcal{L}_s = \sum_{i=1}^{J} \left\|Y_i - X_i\right\|_2, \tag{16}$$

3) Loss function: In our implementation, as same as the base model, the model is supervised at both full sequence scale and single target frame scale. We train the entire network in an end-to-end manner with the total loss:

$$\mathcal{L} = \lambda_f \mathcal{L}_f + \lambda_s \mathcal{L}_s, \tag{17}$$

where λ_f and λ_s are weighting factors.

5. Empirical Experiments and Analyses

5.1. Datasets and Evaluation

The proposed method was trained and evaluated on Human3.6M [8]. The Human3.6M dataset is the most extensive publically available repository of 3D human pose estimation, with 3.6 million images collected from four synchronized cameras running at 50Hz. The 7 individuals featured in the dataset are engaged in 15 different activities, such as ”Waiting”, ”Smoking” and ”Posing”. As per prior research [9], [20], [21], the training set consists of 5 subjects (S1, S5, S6, S7, S8), while S9 and S11 are reserved for evaluation. In this study, we were forced to train on only one-third of the dataset due to computational limitations. The evaluation metric that was used to compare the base model and the modified one is $\mathcal{L}_s = \sum_{i=1}^{J} \|Y_i - X_i\|_2$ .. As we trained our model on a partition of data, we only evaluated it in comparison to the base model, which we trained on the same partition of the data.

5.2. Implementation Details

In this experiment, we used the exact setup of the base model. For the proposed Strided Transformer, the number of encoder layers is set to N1 = N2 = 3, the number of attention heads is h = 8, the embedding dimension is dm = 256 and the hidden units for both VTE and STE are df = 512. The kernel size for the STE layers is kf = 1 and km = 3 and the strided factor is sf = 1 and sm = {3,3,3} for a receptive field of 27 frames. The weighting factors $\mathcal{L}_s$ and $\mathcal{L}_f$ are both set to 1.

5.3. Comparison with State-of-the-art Results

The base method is compared with previous state-of-the-art approaches on Human3.6M dataset [8]. The performance of our 351-frame model with CPN input is reported in Table I. The base method outperforms the state-of-the-art methods on Human3.6M under all metrics (43.7 mm on protocol #1 and 35.2 mm on protocol #2). Two standard evaluation protocols are used in Table I. The mean per joint position error (MPJPE) is the average Euclidean distance between the ground truth and predicted positions of the joints, which is referred to as protocol #1 in many works [22], [23]. Procrustes analysis MPJPE (P-MPJPE) is adopted, where the estimated 3D pose is aligned to the ground truth in translation, rotation, and scale. This protocol is referred to as protocol #2 [9], [10]. Fig. 5 shows some qualitative comparisons with state-of-theart methods [7], which indicates that the base methods can produce more accurate 3D predictions.

5.4. Comparison Between Base and Modified method

To evaluate our changes to model we trained three versions of the base method and compared $\mathcal{L}_s$ on the test data during training. Even though, the modified version's performance is slightly lower than the base method, due to the use of Strided Attention, the modified method is faster than the base method. The comparion is shown in Fig 6. Moreover, As the training was done on Google Colab, we can not provide the exact measurement of speed improvement due to changes in hardware in each runtime. Moreover, two output 3d frame examples are shown in Fig. 7&8.

6. Conclusion

In this work, we investigate ways to improve an existing Transformer-based network for the task of video-based 3D human pose estimation. By changing the attention mechanism and using a sparse attention layer, and making a slight change to the loss function, we managed to decrease the computational cost. Other attention mechanisms and hyperparameter tuning can be investigated for future work to improve the results further.

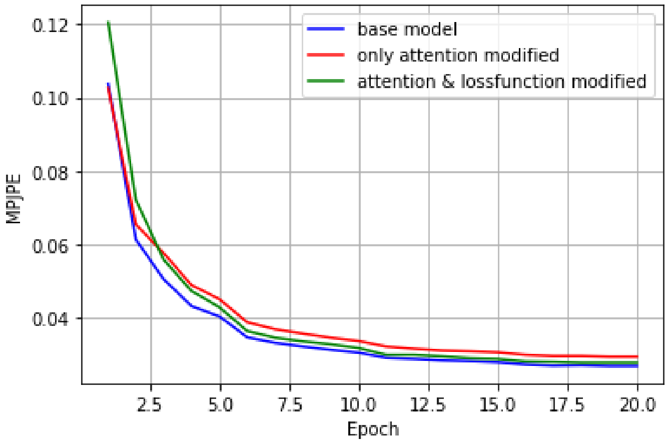


Figure 6. Loss of the test data for three different models.

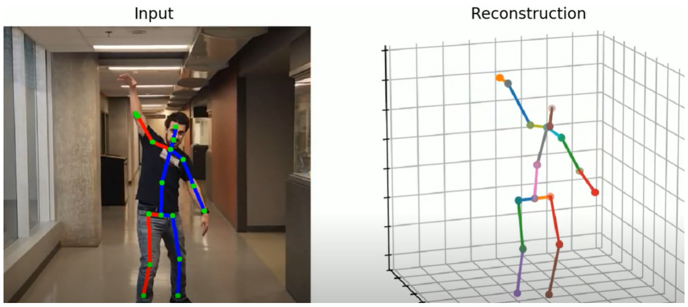


Figure 7. Loss of the test data for three different models.

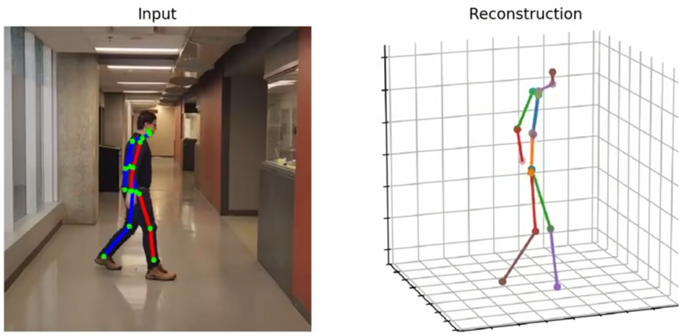


Figure 8. Loss of the test data for three different models.

7. Team members' contributions

The first step of our work was to study some related papers and available public datasets. Each group member was responsible for reviewing two papers, six papers in total. After selecting the best model to work on, each member started looking for methods to tackle and improve the model architecture. Overall, it was decided to mostly work on the attention mechanism and loss function. Amirhosein's focus was mostly on investigating the attention. Mohammadali and Mohammadamin were focused on acquiring improved loss functions for the problem and debugging. For the presentation and preparing the final report, all members worked together to prepare and present the work.

8. Code & Resources

Our code and implementations can be found at https://github.com/hyydrra/StridedTransformer-Pose3D. All the instruction needed to run the network is available in the readme file.

References

[1] I. Radwan, A. Dhall, and R. Goecke, "Monocular image 3d human pose estimation under self-occlusion," in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2013, pp. 1888– 1895.

[2] S. Li and A. B. Chan, "3d human pose estimation from monocular images with deep convolutional neural network," in Asian Conference on Computer Vision (ACCV), 2014, pp. 332–347.

[3] T. Zhao, S. Li, K. N. Ngan, and F. Wu, "3-d reconstruction of human body shape from a single commodity depth camera," IEEE Transactions on Multimedia, vol. 21, no. 1, pp. 114–123, 2018.

[4] P. Hu, E. S.-l. Ho, and A. Munteanu, "3dbodynet: Fast reconstruction of 3d animatable human body shape from a single commodity depth camera," IEEE Transactions on Multimedia, 2021.

[5] A. Kadkhodamohammadi and N. Padoy, "A generalizable approach for multi-view 3d human pose regression," Machine Vision and Applications, vol. 32, no. 1, pp. 1–14, 2021.

[6] K. Pullen and C. Bregler, "Motion capture assisted animation: Texturing and synthesis," in Proceedings of the 29th annual conference on Computer graphics and interactive techniques, 2002, pp. 501–508.

[7] P. Wang, W. Li, Z. Gao, C. Tang, and P. O. Ogunbona, "Depth pooling based large-scale 3-d action recognition with convolutional neural networks," IEEE Transactions on Multimedia, vol. 20, no. 5, pp. 1051–1061, 2018.

[8] M. Garcia-Salguero, J. Gonzalez-Jimenez, and F.-A. Moreno, "Human 3d pose estimation with a tilting camera for social mobile robot interaction," Sensors, vol. 19, no. 22, p. 4943, 2019.

[9] J. Martinez, R. Hossain, J. Romero, and J. J. Little, "A simple yet effective baseline for 3d human pose estimation," in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2640–2649.

[10] K. Lee, I. Lee, and S. Lee, "Propagating lstm: 3d pose estimation based on joint interdependency," in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 119–135.

[11] Y. Cai, L. Ge, J. Liu, J. Cai, T.-J. Cham, J. Yuan, and N. M. Thalmann, "Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks," in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2019, pp. 2272–2281.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in Advances in Neural Information Processing Systems (NIPS), 2017, pp. 5998–6008.

[13] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, "Efficient transformers: A survey," arXiv preprint arXiv:2009.06732, 2020.

[14] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu et al., "A survey on visual transformer," arXiv preprint arXiv:2012.12556, 2020.

[15] M. Geva, R. Schuster, J. Berant, and O. Levy, "Transformer feed-forward layers are key-value memories," arXiv preprint arXiv:2012.14913, 2020.

[16] R. Liu, J. Shen, H. Wang, C. Chen, S.-c. Cheung, and V. Asari, "Attention mechanism exploits temporal contexts: Real-time 3d human pose reconstruction," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 5064–5073.

[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, vol. 36, no. 7, pp. 1325–1339, 2013.

[18] 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, vol. 87, no. 12, pp. 4–27, 2010

[19] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Coarse-to-fine volumetric prediction for single-image 3d human pose," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 7025–7034.

[20] X. Chen, K.-Y. Lin, W. Liu, C. Qian, and L. Lin, "Weakly-supervised discovery of geometry-aware representation for 3d human pose estimation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 10 895–10 904.

[21] D. Tome, M. Toso, L. Agapito, and C. Russell, "Rethinking pose in 3d: Multi-stage refinement and recovery for markerless motion capture," in 2018 International Conference on 3D Vision (3DV), 2018, pp. 474–483.

[22] H.-S. Fang, Y. Xu, W. Wang, X. Liu, and S.-C. Zhu, "Learning pose grammar to encode human body configuration for 3d pose estimation," in Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[23] Shichao Li, Lei Ke, Kevin Pratama, Yu-Wing Tai, Chi-Keung Tang, and Kwang-Ting Cheng. Cascaded deep monocular 3d human pose estimation with evolutionary training data. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, jun 2020

[24] Wenhao Li, Hong Liu, Runwei Ding, Mengyuan Liu, Pichao Wang, and Wenming Yang. Exploiting temporal contexts with strided transformer for 3d human pose estimation, 2021.

[25] Denis Tome, Chris Russell, and Lourdes Agapito. Lifting from the deep: Convolutional 3d pose estimation from a single image. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, jul 2017

[26] Ailing Zeng, Xiao Sun, Fuyang Huang, Minhao Liu, Qiang Xu, and Stephen Lin. Srnet: Improving generalization in 3d human pose estimation with a split-and-recombine approach, 2020.

[27] Jinlu Zhang, Zhigang Tu, Jianyu Yang, Yujin Chen, and Junsong Yuan. Mixste: Seq2seq mixed spatio-temporal encoder for 3d human pose estimation in video, 2022.

[28] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N. Metaxas. Semantic graph convolutional networks for 3d human pose regression. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, jun 2019.

[29] Rewon Child, Scott Gray, Alec Radford, Ilya Sutskever, "Generating Long Sequences with Sparse Transformers", https://arxiv.org/abs/1904.10509, 23 Apr 2019