Uplift and Upsample: Efficient 3D Human Pose Estimation with Uplifting Transformers

Moritz Einfalt Katja Ludwig Rainer Lienhart
Machine Learning and Computer Vision Lab, University of Augsburg
{moritz.einfalt, katja.ludwig, rainer.lienhart}@uni-a.de

Abstract

The state-of-the-art for monocular 3D human pose estimation in videos is dominated by the paradigm of 2D-to-3D pose uplifting. While the uplifting methods themselves are rather efficient, the true computational complexity depends on the per-frame 2D pose estimation. In this paper, we present a Transformer-based pose uplifting scheme that can operate on temporally sparse 2D pose sequences but still produce temporally dense 3D pose estimates. We show how masked token modeling can be utilized for temporal upsampling within Transformer blocks. This allows to decouple the sampling rate of input 2D poses and the target frame rate of the video and drastically decreases the total computational complexity. Additionally, we explore the option of pre-training on large motion capture archives, which has been largely neglected so far. We evaluate our method on two popular benchmark datasets: Human3.6M and MPI-INF-3DHP. With an MPJPE of 45.0 *mm and* 46.9 *mm, respectively, our proposed method can compete with the state-of-the-art while reducing inference time by a factor of 12. This enables real-time throughput with variable consumer hardware in stationary and mobile applications. We release our code and models at* `https://github.com/goldbricklemon/uplift-upsample-3dhpe`

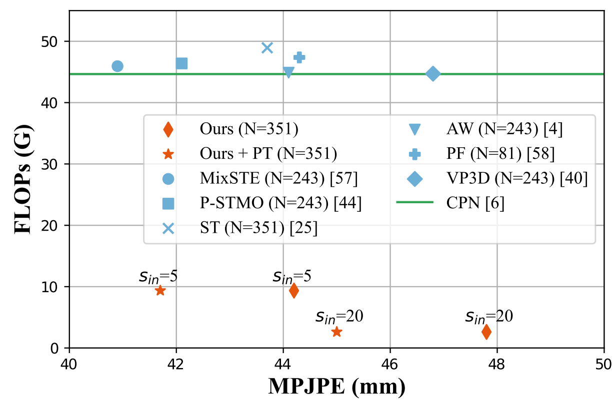

Figure 1: Spatial precision (MPJPE) and per-frame computational complexity (FLOPs) for different pose uplifting methods on Human3.6M (lower is better). The measured FLOPs include the necessary 2D pose estimation, here with CPN [6]. *+PT* denotes pre-training on motion capture data.

1. Introduction

Following the huge advancements in 2D human pose estimation (HPE) over the last years, much research has been dedicated to monocular 3D HPE. Reconstructing the locations of human joints or other body landmarks in 3D space from a single RGB camera has huge potential, with applications in computer animation [39, 38, 42], action recognition [34, 56, 30, 23, 5], or posture and motion analysis [54, 41, 52]. Monocular 3D HPE becomes even more relevant when the complete process can be handled on mobile computers or smartphones. It opens up yet another field of control and interaction applications [12, 18].

Current methods for 3D HPE in videos mainly follow the paradigm of 2D-to-3D pose uplifting [36, 11, 51]. This two-stage approach consistently leads to the highest spatial precision on common 3D HPE benchmarks [19, 37]. It utilizes an existing image-based 2D HPE model to generate 2D poses for every single video frame. Then, a separate uplifting model is trained to estimate the 3D poses for a sequence of frames based on only the sequential 2D pose estimates [2, 40, 28, 55, 53, 49, 16]. Since the 2D poses are a rather compact input description, the uplifting paradigm allows for models that operate on very long input sequences that can span multiple seconds in high frame rate recordings [45, 4, 25, 26]. This is otherwise hardly possible when directly operating on the raw video frames [20, 21]. Ongoing research mainly focuses on further improving the spatial precision of 3D pose estimates. Some recent work also analyzes the computational complexity of the uplifting process itself, but fails to capture the true complexity for possible applications [58, 44, 57]. This comes from the depen-

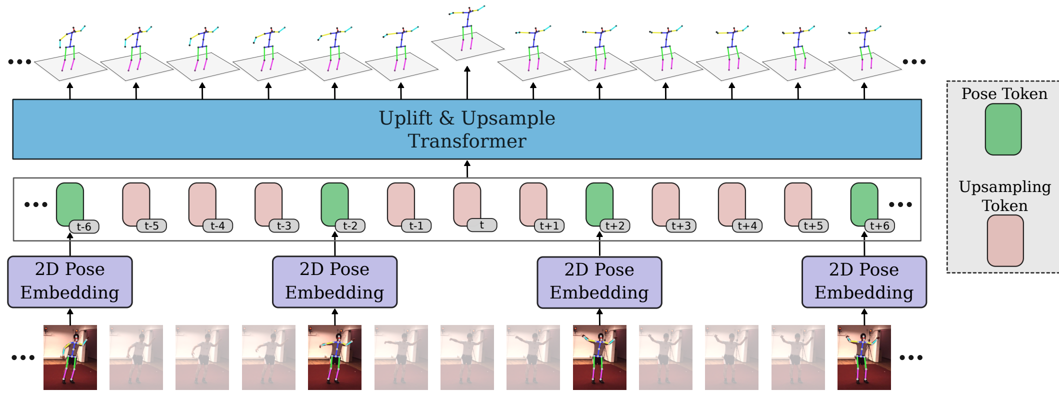

Figure 2: We extract 2D poses at a fixed key-frame interval and transform them into pose tokens. After padding this sequence with a learnable upsampling token, our Transformer network generates dense 3D pose predictions at the target frame rate. During inference, we only use the prediction for the central frame and the entire video is processed in sliding-window fashion.

dence on 2D HPE models with high spatial precision, such as Mask R-CNN [15], CPN [6] or HRNet [7]. The total complexity of 3D HPE via pose uplifting is typically dominated by the initial 2D pose estimation process (see Figure 1). This prohibits applications where real-time throughput on compute- and power-limited consumer hardware is required.

In this paper, our main goal is to reduce the overall complexity by limiting the 2D pose estimation to a small fraction of video frames. Existing uplifting models on 2D pose sequences always have the same rate of input and output poses [40, 4, 58]. In contrast, we present a Transformer-based architecture that can operate on temporally sparse input poses but still generate dense 3D pose sequences at the target frame rate. Inspired by the masked token modeling in Transformer architectures [9, 46, 29, 44], we present a tokenization mechanism that allows to upsample the temporal sequence representation within our uplifting Transformer. Missing 2D pose estimates in the input sequence are replaced by position-aware upsampling tokens. These tokens are jointly transformed into 3D pose estimates for their respective video frames via self-attention over the entire sequence (see Figure 2). This greatly reduces the computational complexity and makes our model more flexible regarding the effective input frame rate in potential applications. In fact, the sampling rate of 2D poses can even be adapted based on the expected or observed motion speed. Since training only require annotated 3D pose sequences but no video recordings, we additionally explore pre-training on large-scale motion capture archives. We evaluate its benefits for our and existing Transformer architectures and show how it can counter the adverse effects of sparse input sequences. In summary, our contributions are: (1) We propose a joint uplifting and upsampling Transformer architecture that can generate temporally dense 3D pose predictions from sparse sequences of 2D poses; (2) We evaluate the effect of Transformer pre-training on motion capture data; (3) We show that our method leads to smoother and more precise 3D pose estimates than direct interpolation on sparse output sequences from competing methods. At the same time, it reduces inference time by at least factor 12 and supports different input frame rates during inference. To the best of our knowledge, this is the first paper to explicitly address efficient 2D-to-3D pose uplifting in videos with a sparse-to-dense prediction scheme as well as direct pre-training on large-scale motion capture data.

2. Related Work

2D-to-3D Pose Uplifting in Videos Recent work on 2D-to-3D pose uplifting in videos either uses temporal convolutional networks (TCN) [1], graph convolutional networks (GCN) [8] or Transformer networks [50]. Pavllo *et al.* [40] introduce a TCN-based uplifting model that can leverage long input sequences and partially labeled data. Extensions for this model focus on attention mechanisms [31] or different pose representations [55, 4]. Cai *et al.* [2] propose a GCN-based method that explicitly models dependencies between locally related human joints over space and time and performs well on short input sequences. This framework is extended with either fixed [53] or input-conditioned non-local dependencies [16]. Most recently, Transformer architectures from the vision domain are adapted for pose uplifting. Zheng *et al.* [58] introduce a Transformer architecture for temporal uplifting with self-attention over space or time. Li *et al.* [25] use a strided Transformer block to handle longer input sequences more efficiently. Zhang *et al.* [57] propose a joint-wise temporal attention block for smaller and more efficient uplifting Transformers. We follow the trend of Transformer-based uplifting and use building blocks from [58, 25] to form a deeper architecture that is designed for sparse input sequences.

Efficient 3D HPE Most existing work on efficient 3D HPE focuses on end-to-end vision models for single images. Mehta *et al.* [39] propose a custom CNN with real-time capabilities on consumer GPUs. Hwang *et al.* [18] use CNN model compression to learn a very efficient model for mobile devices. These methods lack reasoning over multiple video frames, which is crucial for countering the 2D-to-3D ambiguity of monocular 3D pose reconstruction [25, 57]. Mehta *et al.* [38] propose a real-time capable mixed CNN/MLP architecture to predict single-frame 3D poses with additional temporal skeleton fitting. In the scope of temporal 2D-to-3D pose uplifting, most methods are rather efficient with real-time speed on consumer hardware [40, 44, 57]. This, however, does not factor in the computational requirements for the initial per-frame 2D pose estimation. In combination, the uplifting itself accounts for only a fraction of the total complexity. We address this issue and show that pose uplifting on sparse input sequences can easily reach real-time throughput (*i.e.* with some constant delay) while maintaining spatial prediction accuracy.

Data Augmentation and Pre-Training 3D HPE has the common difficulty that datasets with paired video and 3D pose data are scarce and have limited variability in visual appearance and human motion. 2D-to-3D pose uplifting methods have the advantage that paired data is not required [11, 51]. This allows for data augmentation strategies on 3D motion sequences alone. Li *et al.* [24] use evolutionary operators to generate variations of existing 3D poses. Gong *et al.* [14] use a generative model to create plausible new 3D poses. Both methods are limited to single poses, however. Gong *et al.* [13] use a hallucination model to predict a new motion sequence from a given start and end pose, which is then refined by a dynamic skeleton model within a physics simulation. In the scope of pre-training, Transformer architectures are known to benefit from training on large-scale datasets, including Transformers in the vision domain [3, 10, 48]. Shan *et al.* [44] show how an uplifting Transformer can be trained with the pre-text task of 2D pose sequence reconstruction. In contrast, we evaluate the benefits of pre-training on archives of unpaired motion capture data, for our and existing Transformer architectures.

3. Method

Our method follows the common 2D-to-3D pose uplifting approach: Given a video recording V with frame indices $\mathcal{S} = \{1, \ldots, |V|\}$, we use an off-the-shelf 2D HPE model to obtain 2D input poses $\{p_t\}_{t \in \mathcal{S}}$.[1] Each pose $p_t \in \mathbb{R}^{J \times 2}$ is described by the normalized 2D image coordinates [2] of J designated human joints. We discard the detection scores or any other occlusion information that the 2D HPE model might provide. The goal is to recover the actual 3D pose $P_t^{\text{gt}} \in \mathbb{R}^{J \times 3}$, *i.e.* the metric camera coordinates of every joint, for every single video frame. Our main focus lies on using only a subset of input 2D poses at equidistant key-frames $\mathcal{S}_{\text{in}} = \{t | t \bmod s_{\text{in}} = 0\}_{t \in \mathcal{S}}$ with input stride s_{in}. At the same time, we want to generate 3D pose predictions at a higher rate, *i.e.* with a smaller output stride $s_{\text{out}} < s_{\text{in}}$, at frame indices $\mathcal{S}_{\text{out}} = \{t | t \bmod s_{\text{out}} = 0\}_{t \in \mathcal{S}}$. For simplicity, we will assume $s_{\text{in}} = k \cdot s_{\text{out}}$, $k \in \mathbb{N}$, such that $\mathcal{S}_{\text{in}} \subseteq \mathcal{S}_{\text{out}}$. Ideally, with $s_{\text{out}} = 1$, we can predict dense 3D poses at the full frame rate. Figure 2 depicts an example with $\mathcal{S}_{\text{in}} = \{1, 5, 9, \ldots\}$ and $\mathcal{S}_{\text{out}} = \{1, 2, 3, \ldots\}$.

3.1. Uncoupling of Input and Output Sample Rate

Existing uplifting methods have an identical input/output sample rate of 2D/3D poses [40, 4, 58]. Each model is optimized for a fixed input rate. When trained on sub-sampled input poses with stride $s_{\text{in}} > 1$, the model can only predict 3D poses at an equal output stride $s_{\text{out}} = s_{\text{in}}$. To obtain predictions at the full frame rate, we can use bilinear interpolation between 3D poses at adjacent key-frames. This naive method has two drawbacks: First, each model is still optimized for a single, now reduced, frame rate. This lacks flexibility regarding applications with different input frame rates or variable computational resources. Second, using large values for s_{in} will increasingly deteriorate the predictions at the full frame rate. Simple bilinear interpolation cannot reconstruct the actual human motion between two key-frames. Next, we describe our proposed architecture that can circumvent both issues by uncoupling s_{in} and s_{out}.

3.2. Joint Uplifting and Upsampling

Our solution follows the recent trend of 2D pose sequence uplifting with Transformer networks [58]. The self-attentive nature of Transformers has shown to be well suited to leverage relationships between individual joints of a single human pose [27] and within long sequences of poses in time [25, 57]. The main design criterion for our uplifting network is a comparatively deeper architecture that can operate on variably sparse input sequences. It has to concurrently handle pose uplifting and temporal upsampling. At the same time, the model should stay efficient in training and inference. Similar to most temporal pose uplifting methods, our model operates on 2D pose sequences with fixed length $N = 2n + 1$, $n \in \mathbb{N}$. Each sequence covers frame indices $\mathcal{S}_N = \{t - n, \ldots, t + n\}$ around an output frame $t \in \mathcal{S}_{\text{out}}$. Since our Transformer-based model aggregates information from the entire sequence, its temporal

[1] We use this notation to describe an ordered sequence.

[2] We assume known camera intrinsics and map 2D coordinates to $[-1, 1]$ while preserving the aspect ratio.

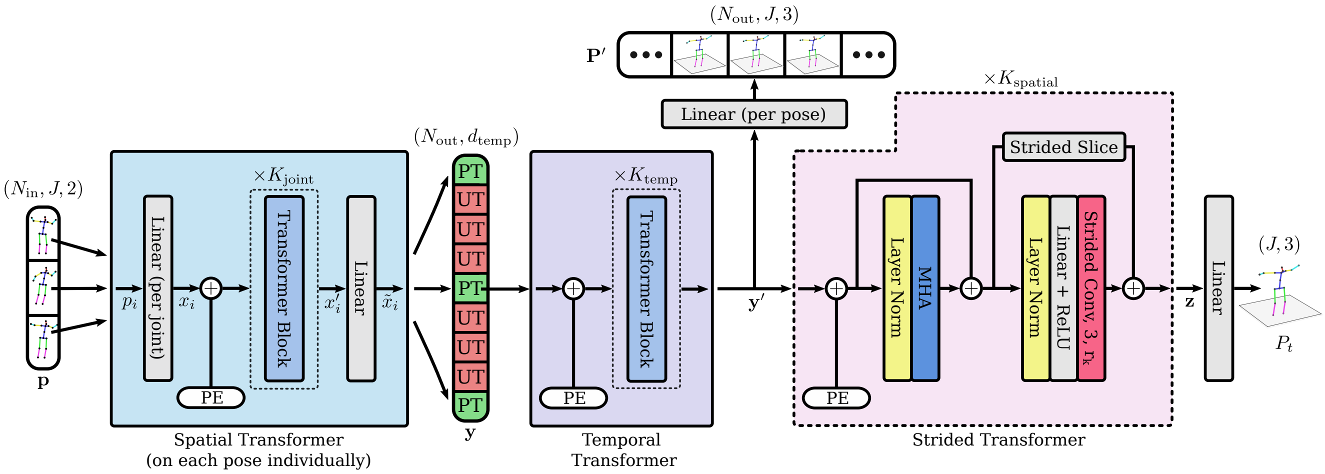

Figure 3: Instantiation of our architecture, with $N = 9$, input stride $s_{\text{in}} = 4$ and output stride $s_{\text{out}} = 1$. The spatial and temporal Transformer use a learnable positional embedding (PE) and vanilla Transformer blocks with multi-head self-attention (MHA). The strided Transformer employs strided convolution with stride r_k and kernel size 3. PT denotes a pose token $\tilde{x}_i$, UT the learnable upsampling token.

receptive field [40] equals N. However, due to key-frame sub-sampling, the actual input to our model is comprised of only the key-frame 2D poses $\mathbf{p}$ with:

$$\mathbf{p} = \{p_i | i \in \mathcal{S}_{\text{in}} \cap \mathcal{S}_N\}. \tag{1}$$

The effective input sequence length is therefore reduced to $N_{\text{in}} := |\mathbf{p}| \leq \frac{N-1}{s_{\text{in}}} + 1$. During training, the model generates intermediate 3D pose predictions $P'_i \in \mathbb{R}^{J \times 3}$ for all output frames within the temporal receptive field:

$$\mathbf{P}' = \{P'_i | i \in \mathcal{S}_{\text{out}} \cap \mathcal{S}_N\}. \tag{2}$$

Additionally, the 3D pose for the central frame t is further refined to its final prediction P_t. During evaluation, only this central prediction is used. We utilize three distinct Transformer sub-networks from recent literature that, in combination, fit our main design goals. Figure 3 provides an overview of our architecture.

Joint-Wise Spatial Transformer The first sub-network is shared between all input poses and operates on each pose individually. It uses self-attention across individual joints to form a strong pose representation for the subsequent sub-networks. Each input pose p_i is first linearly mapped to an initial joint-wise embedding $x_i \in \mathbb{R}^{J \times d_{\text{joint}}}$. After adding a positional embedding to encode the type of each joint, we use K_{joint} spatial Transformer blocks [58] that operate on the sequence of joint embeddings. The output is the joint-aware per-pose embedding $x'_i \in \mathbb{R}^{J \times d_{\text{joint}}}$. It is subsequently condensed into a 1D encoding $\tilde{x}_i \in \mathbb{R}^{d_{\text{temp}}}$. We refer to all $\tilde{x}_i$ as our initial pose tokens.

Pose-Wise Temporal Transformer The second sub-network uses vanilla Transformer blocks with self-attention across the temporal sequence of pose tokens. This building block is a quasi-standard in recent Transformer-based uplifting methods [58, 25, 44]. We extend its usual objective of direct 3D pose reconstruction for key-frame poses within the input sequence. We want to generate smooth and temporally consistent 3D poses for all output frames within the temporal receptive field. We present a simple modification that enables simultaneous uplifting and upsampling within the temporal Transformer blocks. First, we recombine the key-frame pose tokens $\tilde{x}_i$ from the spatial Transformer into a temporal sequence. We then pad this sequence to target length $N_{\text{out}} := |\mathbf{P}'|$. For this, we adopt the masked token modeling of Transformers [29, 44] and introduce an upsampling token $u \in \mathbb{R}^{d_{\text{temp}}}$. It is a learnable parameter that is initialized randomly and optimized during training. This token acts as a placeholder at all non-key-frame indices. Figure 2 depicts this gap-filling process. The elements of the padded token sequence $\mathbf{y} = \{y_i | i \in \mathcal{S}_{\text{out}} \cap \mathcal{S}_N\}$ are then defined as

$$y_i = \begin{cases} \tilde{x}_i & \text{if } i \in \mathcal{S}_{\text{in}}, \\ u & \text{else.} \end{cases} \tag{3}$$

In contrast to [44], the token u not only encodes a pre-text task of input reconstruction, but rather the reconstruction of the upsampled sequence in output space. A second positional embedding ensures that, in particular, each instantiation of the upsampling token is conditioned on its relative frame index. We feed the token sequence to a stack of K_{temp} vanilla Transformer blocks. We restrict any attention within the first Transformer block to the pose tokens, since the initial upsampling tokens do not carry input related information. This is implemented by only computing self-attention keys and values [50] from the pose tokens. The output $\mathbf{y}' \in \mathbb{R}^{N_{\text{out}} \times d_{\text{temp}}}$ after the last Transformer block encodes the uplifted and upsampled 3D poses for all output frames. We use a single linear mapping to obtain the intermediate 3D pose predictions $\mathbf{P}'$.

Sequence Reduction for Single-Frame Output One main drawback of vanilla Transformer blocks is the quadratic complexity w.r.t. the sequence length. Stacking a large number of vanilla Transformer blocks at full temporal resolution does not align with our goal of an overall efficient model. Ultimately, our model is designed for a symmetric N-to-1 prediction scheme during evaluation. This operation mode commonly delivers superior results, as the prediction for the central sequence index is based on an equal amount of past and future information [58, 44]. To further refine the pose prediction specifically for the central index t, it is not necessary to keep the full sequence representation of length N_{out}. Our third sub-network therefore incrementally reduces the previous sequence representation $\mathbf{y}'$, until only the refined output for the central sequence index remains. This allows us to add additional temporal self-attention blocks, but keep the overall complexity feasible. We utilize K_{stride} strided Transformer blocks [25], which use strided convolutions instead of a simple MLP. The details are depicted in Figure 3. Each block k reduces the sequence length by stride factor r_k. We choose all r_k such that the output $\mathbf{z}$ after the last block is $\mathbf{z} \in \mathbb{R}^{1 \times d_{\text{temp}}}$. A single linear mapping generates the final 3D pose prediction $P_t \in \mathbb{R}^{J \times 3}$ for the central sequence index.

Sequence and Center Supervision The entire architecture is trained with two separate objectives. We use the center frame loss $\mathcal{L}_{\text{center}}$ to minimize the root-relative mean per-joint position error (MPJPE) [19] of the refined 3D pose prediction for central target frame t:

$$\mathcal{L}_{\text{center}} = \frac{1}{J} \sum_{j=1}^{J} \|(P_{t,j} - P_{t,r}) - (P_{t,j}^{\text{gt}} - P_{t,r}^{\text{gt}})\|_2, \quad (4)$$

where the pelvis is commonly used as the designated root joint r. Additionally, we define the MPJPE sequence loss $\mathcal{L}_{\text{seq}}$ on the intermediate 3D pose predictions P'_i for the entire upsampled sequence:

$$\mathcal{L}_{\text{seq}} = \frac{1}{J \cdot N_{\text{out}}} \sum_{i \in \mathcal{S}_{\text{out}} \cap \mathcal{S}_N} \sum_{j=1}^{J} \|(P'_{i,j} - P'_{i,r}) - (P_{i,j}^{\text{gt}} - P_{i,r}^{\text{gt}})\|_2. \quad (5)$$

This form of full-sequence supervision encourages temporally stable predictions [25, 57], which is especially important in our setting of sparse input sequences. The total loss is the weighted sum $\alpha_1 \mathcal{L}_{\text{seq}} + \alpha_2 \mathcal{L}_{\text{center}}$.

Training and Inference We optimize our model for a fixed output stride s_{out}, but for multiple input strides s_{in} simultaneously. Thus, it supports different input frame rates, depending on the application and the available hardware. During training, we utilize 3D pose annotations at the full frame rate and generate all possible key-frame sequences around each frame index $t \in \mathcal{S}$. For inference, only the key-frame poses starting at the first video frame are available and we apply our model at every output frame $t \in \mathcal{S}_{\text{out}}$. In case of $s_{\text{out}} > 1$, 3D pose predictions at the full frame rate are obtained via bilinear interpolation. We always evaluate at the full video frame rate for fair comparison.

3.3. MoCap Pre-Training

In order to further unlock the potential of Transformer architectures in 2D-to-3D pose uplifting, we additionally explore the effects of pre-training on large-scale motion capture data. In this work, we utilize the AMASS [35] meta-dataset. It is a collection of a wide variety of existing motion capture datasets with over 60 hours of human motion. The raw motion capture data has been re-targeted to generate a detailed 3D mesh model of the captured person in compact SMPL parameterization [32]. We reduce the mesh to the target set of J joints. Each joint's 3D location is expressed as a weighted linear combination of a small set of mesh vertices. The mixing weights can either be directly optimized on data [22] or created by a small number of hand annotations. Finally, we project the 3D pose sequences to 2D space with randomly selected virtual cameras. For simplicity, we use the same camera parameters from our final target datasets. The resulting 2D-3D pose sequence pairs can then be directly used for training. Note that the 2D poses are perfect projections and thus without errors. Our model will adjust to the error cases of a 2D pose estimation model during subsequent fine-tuning.

4. Experiments

We evaluate our proposed method on two well-known 3D HPE datasets and compare it with the current state-of-the-art in 2D-to-3D pose uplifting. We also conduct a series of ablation experiments to reveal the impact of sparse input sequences, explicit upsampling and large-scale pre-training on spatial accuracy and inference efficiency. Additional experiments on multi-stride training, augmentation strategies and architecture components as well as qualitative examples can be found in the supplementary material.

4.1. Datasets

Human3.6M [19] is the most common dataset for indoor 3D HPE. It consists of 11 actors performing 15 different actions each. They are recorded by four stationary RGB cameras at 50 Hz. We follow the standard evaluation protocol from previous work [36, 11, 40, 58]: Five subjects

Table 1: Results on Human3.6M with CPN [6] 2D poses. We evaluate according to Protocol 1 (MPJPE, top) and Protocol 2 (P-MPJPE, bottom). Best results are bold, second best results are underlined. (*) uses the refinement module from [2]. *+PT* denotes MoCap pre-training on AMASS.

MPJPE (mm)↓	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Pur.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg
Cai *et al.* [2] ICCV'19 (N = 7)(*)	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.* [40] CVPR'19 (N = 243)	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
Xu *et al.* [55] CVPR'20 (N = 9)	**37.4**	43.5	42.7	42.7	46.6	59.7	41.3	45.1	<u>52.7</u>	60.2	45.8	43.1	47.7	33.7	37.1	45.6
Zheng *et al.* [58] ICCV'21 (N = 81)	41.5	44.8	39.8	42.5	46.5	51.6	42.1	42.0	53.3	60.7	45.5	43.3	46.1	31.8	32.2	44.3
Shan *et al.* [45] MM'21 (N = 243)	40.8	44.5	41.4	42.7	46.3	55.6	41.8	41.9	53.7	60.8	45.0	41.5	44.8	30.8	31.9	44.3
Chen *et al.* [4] TCSVT'21 (N = 243)	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
Li *et al.* [25] TMM'22 (N = 351)(*)	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
Hu *et al.* [16] MM'21 (N = 96)	38.0	43.3	39.1	**39.4**	45.8	53.6	41.4	41.4	55.5	61.9	44.6	41.9	44.5	31.6	29.4	43.4
Li *et al.* [26] CVPR'22 (N = 351)	39.2	43.1	40.1	40.9	44.9	51.2	40.6	41.3	53.5	60.3	43.7	41.1	43.8	29.8	30.6	43.0
Shan *et al.* [44] arXiv'22 (N = 243)(*)	38.4	42.1	39.8	40.2	45.2	48.9	<u>40.4</u>	**38.3**	53.8	<u>57.3</u>	43.9	41.6	42.2	29.3	29.3	42.1
Zhang *et al.* [57] CVPR'22 (N = 243)	<u>37.6</u>	**40.9**	**37.3**	<u>39.7</u>	**42.3**	49.9	**40.1**	<u>39.8</u>	**51.7**	**55.0**	**42.1**	**39.8**	**41.0**	**27.9**	**27.9**	**40.9**
Ours (N = 351), $s_{\text{in}} = s_{\text{out}} = 5$	41.8	45.5	41.8	44.2	48.4	54.2	43.7	43.1	58.9	66.3	46.1	43.7	46.0	30.9	31.2	45.7
Ours (N = 351), $s_{\text{in}} = s_{\text{out}} = 5$ (*)	39.6	43.8	40.2	42.4	46.5	53.9	42.3	42.5	55.7	62.3	45.1	43.0	44.7	30.1	30.8	44.2
Ours (N = 351), $s_{\text{in}} = s_{\text{out}} = 5$, + PT	40.6	42.7	38.5	41.1	45.2	<u>48.7</u>	41.5	41.0	53.3	61.3	43.3	<u>41.0</u>	42.3	30.0	<u>29.0</u>	42.6
Ours (N = 351), $s_{\text{in}} = s_{\text{out}} = 5$, + PT (*)	38.6	<u>41.0</u>	<u>37.6</u>	<u>39.7</u>	<u>44.2</u>	**47.9**	40.9	<u>39.8</u>	**51.7**	60.3	<u>43.1</u>	41.1	<u>41.6</u>	<u>28.4</u>	29.2	<u>41.7</u>
Ours (N = 351), $s_{\text{in}} = 20$, $s_{\text{out}} = 5$	45.4	47.9	43.4	47.2	49.6	55.9	46.4	45.4	59.9	66.7	47.5	45.5	49.8	33.0	33.8	47.8
Ours (N = 351), $s_{\text{in}} = 20$, $s_{\text{out}} = 5$, + PT	44.5	45.1	40.3	44.6	46.3	50.7	44.4	43.7	54.6	62.3	44.9	43.1	47.0	32.3	31.9	45.0

P-MPJPE (mm)↓	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Pur.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg
Cai *et al.* [2] ICCV'19 (N = 7)(*)	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
Pavllo *et al.* [40] CVPR'19 (N = 243)	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.* [55] CVPR'20 (N = 9)	31.0	34.8	34.7	34.4	36.2	43.9	31.6	33.5	<u>42.3</u>	49.0	37.1	33.0	39.1	26.9	31.9	36.2
Chen *et al.* [4] TCSVT'21 (N = 243)	33.1	35.3	33.4	35.9	36.1	41.7	32.8	33.3	42.6	49.4	37.0	32.7	36.5	25.5	27.9	35.6
Li *et al.* [25] TMM'22 (N = 351)(*)	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
Li *et al.* [26] CVPR'22 (N = 351)	31.5	34.9	32.8	33.6	35.3	39.6	32.0	32.2	43.5	48.7	36.4	32.6	34.3	23.9	25.1	34.4
Shan *et al.* [45] MM'21 (N = 243)	32.5	36.2	33.2	35.3	35.6	42.1	32.6	31.9	42.6	<u>47.9</u>	36.6	32.1	34.8	24.2	25.8	35.0
Zheng *et al.* [58] ICCV'21 (N = 81)	32.5	34.8	32.6	34.6	35.3	39.5	32.1	32.0	42.8	48.5	<u>34.8</u>	32.4	35.3	24.5	26.0	34.6
Shan *et al.* [44] arXiv'22 (N = 243)	31.3	35.2	32.9	33.9	35.4	39.3	32.5	31.5	44.6	48.2	36.3	32.9	34.4	23.8	23.9	34.4
Hu *et al.* [16] MM'21 (N = 96)	**29.8**	34.4	31.9	**31.5**	35.1	40.0	**30.3**	<u>30.8</u>	42.6	49.0	35.9	<u>31.8</u>	35.0	25.7	<u>23.6</u>	<u>33.8</u>
Zhang *et al.* [57] CVPR'22 (N = 243)	<u>30.8</u>	**33.1**	**30.3**	<u>31.8</u>	**33.1**	<u>39.1</u>	<u>31.1</u>	**30.5**	42.5	**44.5**	**34.0**	**30.8**	**32.7**	**22.1**	**22.9**	**32.6**
Ours (N = 351), $s_{\text{in}} = s_{\text{out}} = 5$ (*)	32.7	36.1	33.4	36.0	36.1	42.0	33.3	33.1	45.4	50.7	37.0	34.1	35.9	24.4	25.4	35.7
Ours (N = 351), $s_{\text{in}} = s_{\text{out}} = 5$, + PT (*)	31.6	<u>33.7</u>	<u>31.8</u>	33.3	<u>34.7</u>	**38.7**	32.2	31.2	**41.9**	48.9	35.5	32.6	<u>33.7</u>	<u>23.4</u>	24.0	<u>33.8</u>
Ours (N = 351), $s_{\text{in}} = 20$, $s_{\text{out}} = 5$	37.6	38.9	36.0	39.4	38.2	44.1	36.4	35.2	48.3	52.9	38.6	35.8	39.6	26.8	27.5	38.4
Ours (N = 351), $s_{\text{in}} = 20$, $s_{\text{out}} = 5$, + PT	36.4	36.5	33.0	37.1	36.4	40.1	34.8	34.3	45.0	50.1	36.9	34.2	37.9	26.5	25.8	36.3

(S1, S5, S6, S7, S8) are used for training, while evaluation is performed on two subjects (S9, S11). We use 2D poses from a fine-tuned CPN [6] during training and evaluation.

MPI-INF-3DHP [37] is a smaller but more challenging dataset for single-person 3D HPE with more diversity in motion, viewpoints and environments. The training data consists of eight actors performing various actions in a green screen studio with 14 RGB cameras. The evaluation data consists of indoor and outdoor recordings of six actors from a single camera. We sample all recordings to a common frame rate of 25 Hz. Since some test-set videos are at 50 Hz, we use additional bilinear upsampling on the estimated 3D poses to evaluate on the full frame rate. We use ground truth 2D poses in all experiments for best comparison with existing work.

Metrics We evaluate results on Human3.6M with the MPJPE metric [19] (Equation 4). We additionally report the N-MPJPE [43] and P-MPJPE [36], *i.e.* the MPJPE after scale or procrustes alignment. Evaluation with MPJPE and P-MPJPE is often referred to as Protocol 1 and 2, respectively. For MPI-INF-3DHP, we report the MPJPE, the percentage of correct keypoints (PCK) at a maximum misplacement of 150 mm and the area under the curve (AUC) with a threshold range of 5-150 mm [37].

4.2. Implementation Details

We instantiate our architecture with $K_{\text{joint}} = 4$, $K_{\text{temp}} = 4$ and $K_{\text{strided}} = 3$ Transformer blocks, with an internal representation size of $d_{\text{joint}} = 32$ and $d_{\text{temp}} = 348$. The spatial and temporal Transformer use stochastic depth [17] with a drop rate of 0.1. We evaluate temporal receptive fields of $N \in \{81, 351\}$. For $N = 351$, we use $s_{\text{out}} = 5$ and train on variable input strides $s_{\text{in}} \in [5, 10, 20]$. For $N = 81$, we use $s_{\text{out}} = 2$ and $s_{\text{in}} \in [4, 10, 20]$. For best results, we extend our architecture with the 3D pose refinement module from [2]. It uses camera intrinsics for reprojection to improve the orientation of some 3D pose estimates. Our model is trained with AdamW [33] for 120 epochs and a batch size of 512. We employ standard data augmentation with horizontal flipping of input poses during training and evalua-

Table 2: MPI-INF-3DHP results on ground truth 2D poses.

Method	PCK↑	AUC↑	MPJPE↓
Pavllo *et al.* [40] CVPR'19 (N = 81)	86.0	51.9	84.0
Chen *et al.* [4] TCSVT'21 (N = 81)	87.9	54.0	78.8
Zheng *et al.* [58] ICCV'21 (N = 9)	88.6	56.4	77.1
Wang *et al.* [53] ECCV'20 (N = 96)	86.9	62.1	68.1
Li *et al.* [26] CVPR'22 (N = 9)	93.8	63.3	58.0
Zhang *et al.* [57] CVPR'22 (N = 27)	94.4	66.5	54.9
Hu *et al.* [16] MM'21 (N = 96)	**97.9**	69.5	42.5
Shan *et al.* [44] arXiv'22 (N = 81)	**97.9**	**75.8**	**32.2**
Ours (N = 81), s_{in} = 10, s_{out} = 2	95.4	67.6	46.9
Ours (N = 81), s_{in} = 10, s_{out} = 2, + PT	+1.7	+2.4	-5.7

Table 3: Results on Human3.6M with $N = 81$ and varying input strides s_{in}. Results are shown for poses on key-frames as well as all frames at 50 Hz.

Method	s_{in}	MPJPE / N-MPJPE / P-MPJPE ↓ Key-frames	MPJPE / N-MPJPE / P-MPJPE ↓ All frames
Strided Transformer [25]	4	49.3 / 47.7 / 38.7	49.4 / 47.8 / 38.7
Pose Former [58]	4	47.7 / 46.3 /37.6	47.7 / 46.3 / 37.6
Ours, $s_{\text{out}} = s_{\text{in}}$	4	**47.6 / 46.0 / 37.3**	47.7 / 46.0 / 37.4
Ours, $s_{\text{out}} = 2$	4	**47.6 / 46.0 / 37.3**	**47.4 / 45.8 / 37.1**
Strided Transformer [25]	10	51.4 / 49.4 / 40.2	52.0 / 50.0 / 40.8
Pose Former [58]	10	48.8 / 46.9 / 38.0	49.3 / 47.4 / 38.5
Ours, $s_{\text{out}} = s_{\text{in}}$	10	**47.5 / 45.8 / 37.1**	48.1 / 46.3 / 37.6
Ours, $s_{\text{out}} = 2$	10	**47.5 / 45.8 / 37.1**	**47.9 / 46.1 / 37.4**
Strided Transformer [25]	20	54.4 / 52.4 / 42.2	57.7 / 55.7 / 45.4
Pose Former [58]	20	48.6 / 47.2 / 38.2	52.0 / 50.6 / 41.4
Ours, $s_{\text{out}} = s_{\text{in}}$	20	**48.1 / 46.4 / 37.6**	51.6 / 49.6 / 40.8
Ours, $s_{\text{out}} = 2$	20	**48.1 / 46.4 / 37.6**	**49.9 / 48.1 / 39.2**

tion. Specifically, we use within-batch augmentation, where the second half of each mini-batch is the flip-augmented version of the first half. We use an initial learning rate of 4e^{-5}, with an exponential decay by 0.99 per epoch. The same schedule is applied to the initial weight decay of 4e^{-6}. The loss weights are fixed at $\alpha_1 = \alpha_2 = 0.5$. All experiments are conducted on a single NVIDIA A100 GPU.

4.3. Results

We compare our method against recent work and the current state-of-the-art. Note that all comparative results use 2D poses at the full frame rate. Table 1 shows the results on Human3.6M. We first evaluate our architecture with a key-frame stride of $s_{\text{in}} = 5$ and no internal upsampling ($s_{\text{in}} = s_{\text{out}}$). The full-rate 3D poses are obtained with bilinear upsampling. Starting at a base MPJPE of 45.7 mm, additional reprojection refinement improves spatial accuracy to 44.2 mm. We can see that our architecture can generate competitive results despite requiring 5 times less input poses. When adding MoCap pre-training (*+PT*), we can further improve results by 2–3 mm. Similar behaviour can be seen for P-MPJPE results. Since we use additional data in this experiment, we do not claim our architecture being better than existing ones. It simply reveals that pre-training can easily compensate the reduced rate of input poses. In order to further reduce the input rate of 2D poses for even larger efficiency gains, we make use of our joint uplifting and upsampling mechanism. With an input stride of $s_{\text{in}} = 20$, we achieve a base MPJPE of 47.8 mm. Reducing 2D poses to only 2.5 Hz thus leads to an increase by ~ 2 mm in MPJPE. But again, with additional pre-training, we can reduce this negative effect to a large extent and obtain competitive results with 45.0 mm MPJPE. At the same time, we are 20 times more efficient in the expensive but required 2D pose estimation and only require a fifth of the forward passes for the uplifting model.

Table 2 shows our results on MPI-INF-3DHP. Our method is even closer to the state-of-the-art on this more challenging dataset. With a setting of $s_{\text{in}} = 10$ and recordings in 25 Hz, we again only require input poses at 2.5 Hz. Despite this huge reduction in complexity, we are able to achieve the currently third best PCK, AUC and MPJPE with 95.4, 67.6 and 46.9 mm, respectively. This confirms the competitiveness of our method despite the constraint of sparse input pose sequences. Additional MoCap pre-training leads to further improvement of 2.4 (AUC) and 5.7 mm (MPJPE). Thus, independent of the target dataset, additional pre-training can reliably improve the spatial precision of 3D pose estimates from sparse 2D poses.

4.4. Ablation Study

We additionally explore how sparse input sequences and MoCap pre-training individually influence our architecture compared to existing uplifting Transformers with similar building blocks. Here, we use a smaller receptive field of $N = 81$ and no refinement module [2]. For easier ablation, we adjust the training recipe of our model as well as the original recipes for the comparative methods in [58, 25] by a small set of common changes: We adopt a batch size of 256 and train for the full number of reported epochs without early stopping. We additionally use an exponentially moving average of the model weights [47] to reduce fluctuations in evaluation results.

Sparse Input Sequences Table 3 shows results on Human3.6M with varying input strides s_{in}. For Pose Former [58] and Strided Transformer [25], where $s_{\text{out}} = s_{\text{in}}$, we adopt bilinear interpolation between output poses at adjacent key-frames to obtain the full-rate 3D poses. We evaluate our model with bilinear ($s_{\text{out}} = s_{\text{in}}$) and learned ($s_{\text{out}} = 2$) upsampling. At a moderate input stride of $s_{\text{in}} = 4$, we observe no difference in prediction quality be-

Table 4: Computational complexity in contrast to best MPJPE on Human3.6M. FLOPs are reported for a single forward pass of the uplifting model. We also report the poses per second (*PPS*) for a video frame rate of 50 Hz on an NVIDIA 1080Ti.

Method	# Params	FLOPs↓	PPS (w/o CPN) ↑	PPS (w/ CPN) ↑	MPJPE (mm)↓
Strided Transformer [25] (N = 351)	4.34 M	2142 M	208	32	**43.7**
Pose Former [58] (N = 81)	9.60 M	1358 M	248	33	44.3
Ours (N = 81), s_{in} = 10, s_{out} = 2, + PT	10.36 M	**543 M**	334	179	45.5
Ours (N = 351), s_{in} = 20, s_{out} = 5, + PT	10.39 M	966 M	**827**	**399**	45.0

Table 5: Results on Human3.6M with $N = 81$, with and without pre-training (*PT*) on AMASS.

Method	#Params	MPJPE / N-MPJPE / P-MPJPE ↓ w/o PT	w/ PT
Strided Transformer [25]	4.06 M	48.1 / 46.6 / 37.7	47.7 / 46.2 / 37.5
Pose Former [58]	9.60 M	**47.4** / 46.0 / 37.4	46.0 / 44.5 / 36.1
Ours, $s_{\text{in}} = s_{\text{out}} = 2$	10.36 M	47.5 / **45.4** / **36.8**	**45.7** / **44.2** / **35.8**

tween full frame rate poses and key-frame poses for all three architectures. With increasing input stride, Strided Transformer results deteriorate notably in both key-frame and all-frame performance. It shows that this architecture is only suitable for long and dense input sequences. Pose Former shows more stable key-frame results, but the full frame rate predictions increasingly suffer from bilinear interpolation. Our architecture, as a deeper combination of the former two, achieves lower spatial precision loss on key-frames with increasing input stride. This advantage carries over to full frame rate results, but pure bilinear interpolation stays a limiting factor for high input strides ($s_{\text{in}} = 20$). Finally, our explicit Transformer-based upsampling leads to a notably smaller gap between key-frame and all-frame performance on all metrics. It is better suited for temporally consistent, full frame rate 3D HPE on sparse input sequences. At the same time, we have a single flexible model that supports different input rates of 2D poses. Existing methods, including Pose Former and Strided Transformer, require a separate model for each input rate.

MoCap Pre-Training Table 5 depicts results on Human3.6M, with and without MoCap pre-training on AMASS. In this experiment we assume dense input sequences. We compare the direct influence of pre-training on the different Transformer-based architectures. We observe that all three uplifting Transformer architectures can improve on the target dataset with additional pre-training. Strided Transformer, with its much lower network capacity, shows only marginal gains. Our architecture achieves a similar MPJPE on pure Human3.6M training compared to Pose Former, but improves the most from pre-training. This shows the benefits of the deeper architecture at an otherwise comparable number of parameters.

Computational Complexity Finally, we compare the computational complexity of our method in Table 4. When looking at the uplifting alone, a single forward pass with $N = 351$ requires similar FLOPs compared to Pose Former and its much smaller sequence length. Our uplifting model requires a forward pass for every s_{out}-th frame only, however. Since the architecture is deeper and thus has more sequential computations, we achieve an uplifting speed-up of roughly $\times 4$ with $s_{\text{out}} = 5$. When we factor in the computational complexity of the a-priori 2D pose estimation, the true gain in efficiency becomes apparent (see also Figure 1). Using CPN to generate input 2D poses, Pose Former and Strided Transformer can no longer meet real-time throughput. Since our model only requires input poses for every s_{in}-th frame, we observe a total speed-up of $\times 12$ for $s_{\text{in}} = 20$, such that 50 Hz videos can easily be processed with real-time throughput on a mid-range consumer GPU. This estimate in speed-up is still rather pessimistic, since we measure CPN inference time for tight image crops. The overhead for per-frame person detection is not included here. The more 2D pose estimation dominates the total complexity, the more the speed-up factor will converge towards s_{in}.

5. Conclusion

We presented a Transformer-based 2D-to-3D pose uplifting model for efficient 3D human pose estimation in videos. It uses building blocks from existing uplifting Transformers to form a deeper architecture that operates on sparse input sequences. With a joint uplifting and upsampling Transformer module, the sparse 2D poses are translated into dense 3D pose predictions. This reduces the overall complexity but allows for temporally consistent 3D poses at the target frame rate of the video. At the same time, we can train and deploy a single flexible model that can operate on different input frame rates. The adverse effects of sparse input sequences can be greatly reduced with pre-training on large-scale motion capture archives. Our experiments reveal a speed-up by factor 12 while still being competitive in spatial precision with current state-of-the art methods.

References

[1] 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.

[2] Yujun Cai, Liuhao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, and Nadia Magnenat Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2272–2281, 2019.

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.

[4] Tianlang Chen, Chen Fang, Xiaohui Shen, Yiheng Zhu, Zhili Chen, and Jiebo Luo. Anatomy-aware 3d human pose estimation with bone-based pose decomposition. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[5] Tailin Chen, Desen Zhou, Jian Wang, Shidong Wang, Yu Guan, Xuming He, and Errui Ding. Learning multi-granular spatio-temporal graph network for skeleton-based action recognition. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 4334–4342, 2021.

[6] 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.

[7] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S Huang, and Lei Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5386–5395, 2020.

[8] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[11] Dylan Drover, Rohith MV, Ching-Hang Chen, Amit Agrawal, Ambrish Tyagi, and Cong Phuoc Huynh. Can 3D Pose be Learned from 2D Projections Alone? In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, September 2018.

[12] Mercedes Garcia-Salguero, Javier Gonzalez-Jimenez, and Francisco-Angel Moreno. Human 3d pose estimation with a tilting camera for social mobile robot interaction. *Sensors*, 19(22):4943, 2019.

[13] Kehong Gong, Bingbing Li, Jianfeng Zhang, Tao Wang, Jing Huang, Michael Bi Mi, Jiashi Feng, and Xinchao Wang. PoseTriplet: Co-evolving 3D human pose estimation, imitation, and hallucination under self-supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11017–11027, June 2022.

[14] Kehong Gong, Jianfeng Zhang, and Jiashi Feng. Poseaug: A differentiable pose augmentation framework for 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8575–8584, 2021.

[15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[16] Wenbo Hu, Changgong Zhang, Fangneng Zhan, Lei Zhang, and Tien-Tsin Wong. Conditional directed graph convolution for 3d human pose estimation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 602–611, 2021.

[17] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European conference on computer vision*, pages 646–661. Springer, 2016.

[18] Dong-Hyun Hwang, Suntae Kim, Nicolas Monet, Hideki Koike, and Soonmin Bae. Lightweight 3d human pose estimation network training using teacher-student learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 479–488, 2020.

[19] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, Jul 2014.

[20] Angjoo Kanazawa, Jason Y Zhang, Panna Felsen, and Jitendra Malik. Learning 3d human dynamics from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5614–5623, 2019.

[21] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. VIBE: Video inference for human body pose and shape estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5253–5263, 2020.

[22] Nikos Kolotouros, Georgios Pavlakos, Michael J. Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[23] Ce Li, Chunyu Xie, Baochang Zhang, Jungong Han, Xiantong Zhen, and Jie Chen. Memory attention networks for skeleton-based action recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[24] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[25] Wenhao Li, Hong Liu, Runwei Ding, Mengyuan Liu, Pichao Wang, and Wenming Yang. Exploiting temporal contexts with strided transformer for 3d human pose estimation. *IEEE Transactions on Multimedia*, 2022.

[26] Wenhao Li, Hong Liu, Hao Tang, Pichao Wang, and Luc Van Gool. Mhformer: Multi-hypothesis transformer for 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13147–13156, June 2022.

[27] Yanjie Li, Shoukui Zhang, Zhicheng Wang, Sen Yang, Wankou Yang, Shu-Tao Xia, and Erjin Zhou. Tokenpose: Learning keypoint tokens for human pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11313–11322, October 2021.

[28] Jiahao Lin and Gim Hee Lee. Trajectory space factorization for deep video-based 3d human pose estimation. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2019.

[29] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1954–1963, 2021.

[30] Mengyuan Liu, Hong Liu, and Chen Chen. Robust 3d action recognition through sampling local appearances and global distributions. *IEEE Transactions on Multimedia*, 20(8):1932–1947, 2017.

[31] Ruixu Liu, Ju Shen, He Wang, Chen Chen, Sen-ching Cheung, and Vijayan Asari. Attention mechanism exploits temporal contexts: Real-time 3d human pose reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5064–5073, 2020.

[32] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. SMPL: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015.

[33] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[34] Diogo C. Luvizon, David Picard, and Hedi Tabia. 2D/3D Pose Estimation and Action Recognition Using Multitask Deep Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

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

[36] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2640–2649, 2017.

[37] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3D Human Pose Estimation In The Wild Using Improved CNN Supervision. *2017 International Conference on 3D Vision (3DV)*, Oct. 2017.

[38] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Mohamed Elgharib, Pascal Fua, Hans-Peter Seidel, Helge Rhodin, Gerard Pons-Moll, and Christian Theobalt. XNect: Real-time multi-person 3D motion capture with a single RGB camera. *ACM Transactions On Graphics (TOG)*, 39(4):82–1, 2020.

[39] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. VNect: Real-time 3D Human Pose Estimation with a Single RGB Camera. *ACM Transactions on Graphics*, 36(4), July 2017.

[40] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3D Human Pose Estimation in Video With Temporal Convolutions and Semi-Supervised Training. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[41] Oky Dicky Ardiansyah Prima, Takashi Imabuchi, Yuta Ono, Yoshitoshi Murata, Hisayoshi Ito, and Yukihide Nishimura. Single camera 3d human pose estimation for tele-rehabilitation. In *eTELEMED 2019: The Eleventh International Conference on eHealth, Telemedicine, and Social Medicine*, pages 13–18, 2019.

[42] Katherine Pullen and Christoph Bregler. Motion capture assisted animation: Texturing and synthesis. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 501–508, 2002.

[43] Helge Rhodin, Mathieu Salzmann, and Pascal Fua. Unsupervised geometry-aware representation for 3d human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[44] Wenkang Shan, Zhenhua Liu, Xinfeng Zhang, Shanshe Wang, Siwei Ma, and Wen Gao. P-stmo: Pre-trained spatial temporal many-to-one model for 3d human pose estimation. *arXiv preprint arXiv:2203.07628*, 2022.

[45] Wenkang Shan, Haopeng Lu, Shanshe Wang, Xinfeng Zhang, and Wen Gao. Improving robustness and accuracy via relative information encoding in 3d human pose estimation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3446–3454, 2021.

[46] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. In *International Conference on Learning Representations*, 2020.

[47] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[48] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.

[49] Shashank Tripathi, Siddhant Ranade, Ambrish Tyagi, and Amit Agrawal. Posenet3d: Learning temporally consistent 3d human pose via knowledge distillation. In *2020 Inter-*

national Conference on 3D Vision (3DV), pages 311–321. IEEE, 2020.

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[51] Bastian Wandt and Bodo Rosenhahn. RepNet: Weakly Supervised Training of an Adversarial Reprojection Network for 3D Human Pose Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[52] Jianbo Wang, Kai Qiu, Houwen Peng, Jianlong Fu, and Jianke Zhu. Ai coach: Deep human pose estimation and analysis for personalized athletic training assistance. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 374–382, 2019.

[53] Jingbo Wang, Sijie Yan, Yuanjun Xiong, and Dahua Lin. Motion guided 3d pose estimation from videos. In *European Conference on Computer Vision*, pages 764–780. Springer, 2020.

[54] Qingqiang Wu, Guanghua Xu, Sicong Zhang, Yu Li, and Fan Wei. Human 3d pose estimation in a lying position by rgb-d images for medical diagnosis and rehabilitation. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*, pages 5802–5805, 2020.

[55] Jingwei Xu, Zhenbo Yu, Bingbing Ni, Jiancheng Yang, Xiaokang Yang, and Wenjun Zhang. Deep kinematics analysis for monocular 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 899–908, 2020.

[56] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[57] Jinlu Zhang, Zhigang Tu, Jianyu Yang, Yujin Chen, and Junsong Yuan. Mixste: Seq2seq mixed spatio-temporal encoder for 3d human pose estimation in video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13232–13242, June 2022.

[58] Ce Zheng, Sijie Zhu, Matias Mendieta, Taojiannan Yang, Chen Chen, and Zhengming Ding. 3d human pose estimation with spatial and temporal transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11656–11665, 2021.