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

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

In the following, we provide additional experiments and qualitative results to further validate our design choices. This includes experiments on upsampling token attention, multi-stride training, within-batch augmentation and architecture components. We additionally discuss limitations of our method and areas of future work.

1. Multi-Stride Training

By default, all our models are trained on multiple input strides s_{in} simultaneously. We first take a closer look at the effects of multi-stride training.

1.1. Single- vs. Multi-Stride Models

Multi-stride training has the benefit of a more flexible model that can operate on different input sample rates of 2D poses. We evaluate, whether this flexibility comes at the cost of reduced spatial precision. Table 1 compares a multi-stride model to separate single-stride models that are trained on one specific input stride $s_{\text{in}} = s_{\text{out}}$ each. The results on Human3.6M [1] show that multi-stride training even improves 3D pose estimates. For small input strides, single and multi-stride models are on par and show no real advantage of either training mode regarding spatial accuracy. With increasing s_{in}, multi-stride training can consistently outperform the single-stride counterpart, on key-frame poses as well as all-frame results. Thus, when aiming for very efficient operation with long input strides, multi-stride training leads to better uplifting and upsampling in 3D output space.

Table 1: Comparison of single- (*SS*) and multi-stride (*MS*) training on Human3.6M. All models are trained with $N = 81$ and $s_{\text{out}} = 2$. The MS models are trained on all input strides $s_{\text{in}} \in \{4, 10, 20\}$ simultaneously. By default, models use deferred upsampling token attention (*DUTA*) within the temporal Transformer. Results are reported for poses on key-frames as well as all frames at 50 Hz.

	s_{in}	MPJPE / N-MPJPE / P-MPJPE ↓ Key-frames	 All frames
SS, w/o DUTA	4	47.9 / **45.8 / 37.1**	47.9 / **45.8 / 37.1**
SS	4	47.9 / **45.8 / 37.1**	47.9 / **45.8 / 37.1**
MS, w/o DUTA	4	59.7 / 54.0 / 43.8	52.7 / 49.0 / 39.6
MS	4	**47.6** / 46.0 / 37.3	**47.4 / 45.8 / 37.1**
SS, w/o DUTA	10	48.0 / 46.1 / 37.4	48.2 / 46.3 / 37.6
SS	10	47.8 / 45.9 / **37.1**	48.0 / **46.1 / 37.3**
MS, w/o DUTA	10	49.9 / 47.7 / 38.6	48.2 / 46.3 / 37.5
MS	10	**47.5 / 45.8 / 37.1**	**47.9 / 46.1** / 37.4
SS, w/o DUTA	20	49.3 / 47.1 / 38.1	50.6 / 48.4 / 39.3
SS	20	50.1 / 47.4 / 38.4	51.5 / 48.8 / 39.6
MS, w/o DUTA	20	49.4 / 47.4 / 38.5	52.1 / 49.8 / 40.5
MS	20	**48.2 / 46.4 / 37.6**	**49.9 / 48.1 / 39.2**

1.2. Upsampling Token Attention

At the beginning of the very first temporal Transformer block, none of the upsampling tokens carry any input-related information. They are only conditioned on their relative frame index. Therefore, any attention to the upsampling tokens will not lead to meaningful information exchange or gain. In contrast, there is even the risk of deteriorating the information carried by the actual pose tokens. To counter this effect, we only allow attention to upsampling tokens from the second temporal Transformer block onward. At this stage, all tokens carry input-related information to some degree. We refer to this as deferred upsampling token attention (DUTA). Table 1 shows results for our multi-stride model on Human3.6M, with and without DUTA. The results clearly show the necessity for DUTA, as it outperforms the vanilla variant with unconstrained cross-token attention on all inputs strides and metrics. The most notable difference occurs when evaluating with $s_{\text{in}} = 4$. In this setting, training without DUTA leads to worse key-frame performance compared to all-frame results. This shows that the pose token representation heavily suffers from unconstrained attention within the first temporal Transformer block. The negative effects are less severe

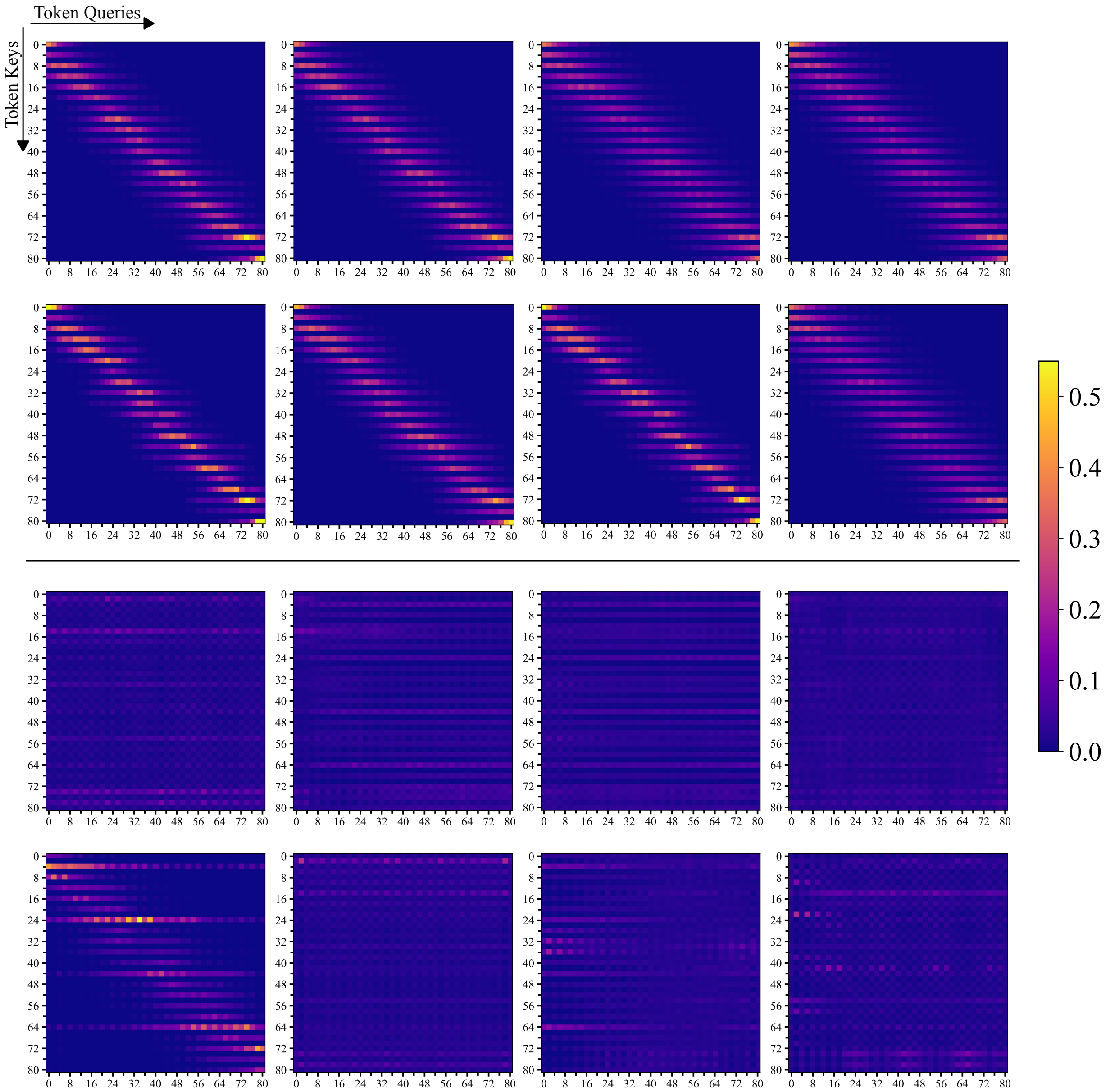

Figure 1: Token attention of the eight MHA heads in the first temporal Transformer block. Top: With DUTA. Bottom: Without DUTA. The corresponding models are trained with $N = 81$, $s_{\text{out}} = 2$ and $s_{\text{in}} \in \{4, 10, 20\}$. The examples are generated with $s_{\text{in}} = 4$, *i.e.* with a ratio of pose and upsampling tokens of one.

for larger s_{in}, but there is still a clear performance gap to the DUTA variant. Thus, delaying the full cross-token attention is a crucial design choice for stable results over different input strides. Table 1 also shows the influence of DUTA on single-stride models. For single-stride training, the ratio between pose and upsampling tokens stays the same for all training examples. The results reveal no clear advantage or disadvantage when training with DUTA in this setting. This shows that DUTA is only required for a variable ratio of pose and upsampling tokens with multi-stride training.

Figure 1 depicts exemplary token attention within the first temporal Transformer block of a multi-stride model. With DUTA, temporal attention shows reasonable token-local information aggregation as often seen in the early stages of a temporal Transformer. Without DUTA, temporal attention is uniformly spread over a distant subset of pose tokens as well as intermediate upsampling tokens. This seems to greatly hinder proper information exchange over the temporal sequence.

Table 2: Effects of within-batch augmentation (*WBA*) on Human3.6M with $N = 351$, $s_{\text{out}} = 5$, $s_{\text{in}} \in \{5, 10, 20\}$ and batch size 512. Results are shown with and without pre-training on motion capture sequences from AMASS.

WBA	s_{in}	MPJPE / N-MPJPE / P-MPJPE ↓ w/o PT	w/ PT
✗	5	46.0 / 44.4 / 36.5	43.5 / 42.1 / 34.7
✓	5	**45.7 / 44.3 / 36.4**	**42.6 / 41.5 / 34.2**
✗	20	48.2 / 46.7 / 38.6	45.7 / 44.4 / 36.8
✓	20	**47.8 / 46.4 / 38.4**	**45.0 / 44.0 / 36.3**

2. Within-Batch Augmentation

Next, we evaluate the benefits of within-batch augmentation (WBA). With WBA, each mini-batch contains the flip-augmented and non-augmented version of a training example. This promotes invariance to horizontal flipping within each weight update. Table 2 shows the results on Human3.6M. WBA leads to a slight performance gain in all metrics, independent of the input stride. Note that this benefit comes at no additional cost during training, since all models in this comparison use a fixed batch size of 512. The result shows that the benefits of WBA outweigh the effectively reduced variability within each mini-batch. We also evaluate with additional pre-training on AMASS. This setting reveals a significant boost through WBA, with a reduction of up to 0.9 mm in MPJPE. We observe that WBA leads to slightly slower convergence during pre-training, but far better validation results. This advantage is then translated over to the fine-tuning on Human3.6M. We also observe similar benefits when experimenting with Pose Former and, to a lesser extent, Strided Transformer.

3. Architecture Components

We also evaluate the individual influence of the three main components of our Transformer architecture: The joint-wise spatial Transformer, the pose-wise temporal Transformer and the strided Transformer. Table 3 compares our full architecture to variants where one component is removed at a time. Starting with the temporal Transformer, this component is the most crucial part of our and related architectures [2, 5]. Removing this block disables repeated self-attention across the entire sequence of pose and upsampling tokens. Additionally, it impedes the full sequence loss $\mathcal{L}_{\text{seq}}$. In combination, the results show that the temporal Transformer is a strict requirement for our architecture to operate properly. Removing the spatial Transformer is less impactful, but we observe a clear drop in precision across all input strides. Thus, dedicating a separate Transformer to generate an initial pose representation is beneficial, especially when input 2D poses are temporally sparse. Finally, the strided Transformer has the lowest impact compared to the other two components, but its removal still leads to an increase in MPJPE by 0.6 - 0.9 mm. It acts as a refinement component via the center frame loss $\mathcal{L}_{\text{center}}$ and is again most helpful for large input strides. Due to the internal temporal striding, it is computationally less expensive compared to the full temporal Transformer blocks and therefore a valid addition to our architecture.

Table 3: MPJPE (mm) on Human3.6M with $N = 81$, $s_{\text{out}} = 2$ and $s_{\text{in}} \in \{4, 10, 20\}$. We compare variants of our architecture with either the spatial Transformer (SPT), the temporal Transformer (TT) or the strided Transformer (ST) removed. The MPJPE is reported relative to the full architecture results.

SPT	TT	ST	$s_{\text{in}} = 4$	$s_{\text{in}} = 10$	$s_{\text{in}} = 20$
✓	✓	✓	**47.4**	**47.9**	**49.9**
	✓	✓	$+1.2$	$+1.2$	$+1.5$
✓		✓	$+4.2$	$+4.7$	$+5.9$
✓	✓		$+0.6$	$+0.7$	$+0.9$

4. Qualitative Examples

Figure 2 depicts qualitative examples on Human3.6M with our method, Pose Former and Strided Transformer at an input stride of $s_{\text{in}} = 20$. In comparison to Strided Transformer, our method typically leads to more precise 3D estimates on key-frames an non-key-frames. Pose Former is more robust to sparse input sequences, but our method still leads to better results on human motion at non-key-frames, *e.g.* during walking motion. Figure 5 depicts additional examples from our best models and 2D input poses at 2.5 Hz. We observe plausible human motion even on difficult examples within Human3.6M and MPI-INF-3DHP. The examples in rows three and six depict failure cases, which we discuss in detail next.

5. Error Modes and Limitations

We observe two main error modes for our proposed method. The first cause of erroneous 3D pose estimates are missdetections within the 2D pose estimates. Figure 5 (rows three, left) depicts such an example, where the estimated 2D locations of leg joints suffer from self-occlusion. Note that the dependence on high quality 2D poses is common to all uplifting methods [4, 6, 2]. Therefore, we see this error mode as a limitation of single-frame 2D HPE and the 2D-to-3D pose uplifting approach in general.

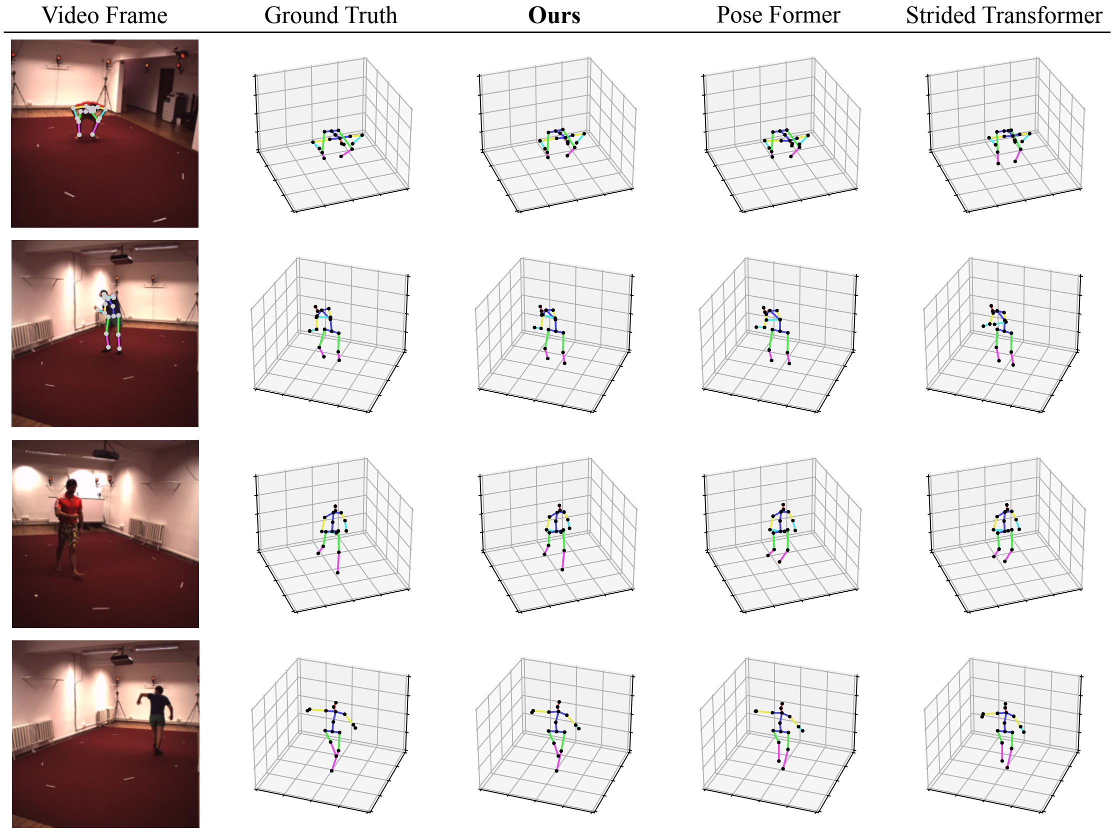

Figure 2: Qualitative examples on Human3.6M, with $N = 81$ and $s_{\text{in}} = 20$. We compare our method, Pose Former [7] and Strided Transformer [2]. Top: Examples on key-frames. Bottom: Examples on non-key-frames.

The second error mode is more unique to our method. Figure 5 (row six, right) depicts an example of a person performing boxing punches in quick succession. We observe that some of the punches are not reconstructed within the estimated 3D pose sequence. Since we utilize input 2D poses at only 2.5 Hz for this example, some of the punches occur so fast that they are completely missing from the subsampled input sequence as well. Consequently, our model is not able to reconstruct the full motion.

In order to analyze the dependency between 2D pose subsampling and fast body motion, we define the average root-relative velocity v_t of a pose P_t as

$$v_t = \frac{1}{J}\sum_{j=1}^{J}\|(P_{t,j} - P_{t,r}) - (P_{t-1,j} - P_{t-1,r})\|_2, \tag{1}$$

where again the pelvis is used as the root joint r. We use the relative velocity, since we focus on the speed of within-body movement. We want to measure fast movement of *e.g.* the arms or legs independent of the person standing in place or walking. Figure 3 shows the MPJPE on Human3.6M in contrast to the ground truth velocity v_t^{gt} (reported in m/s). We divide the range of observable velocities into equidistant intervals and report the MPJPE for all estimated 3D poses within an interval. The results are depicted for the same model under two different settings: a moderate input stride of $s_{\text{in}} = 5$ for high spatial precision and a long input stride of $s_{\text{in}} = 20$ for best efficiency. Under no or very small movement (< 0.2 m/s), both settings perform equally. Due to the limited motion, the temporal component of the input sequence does not offer additional information, no matter the input stride. For movement in the range of 0.2 - 0.4 m/s (*e.g.* walking), both settings show rather stable results, with only minor losses in precision for $s_{\text{in}} = 20$. Most actions within Human3.6M fall into this range of relative pose velocity. Only for considerably faster movement, the results of both settings diverge. While $s_{\text{in}} = 5$ (*i.e.* 2D poses at 10 Hz) stays relatively stable around 50 mm MPJPE, our fastest setting with $s_{\text{in}} = 20$ shows increasing difficulties in reconstructing the true pose sequence in 3D space. This reveals the main limitation of our method: The choice of efficiency, which is mainly governed by s_{in}, must not only fit potential hardware and runtime requirements, but also the range of expected movement speed. While our most efficient setting with $s_{\text{in}} = 20$ is suitable for regular movement

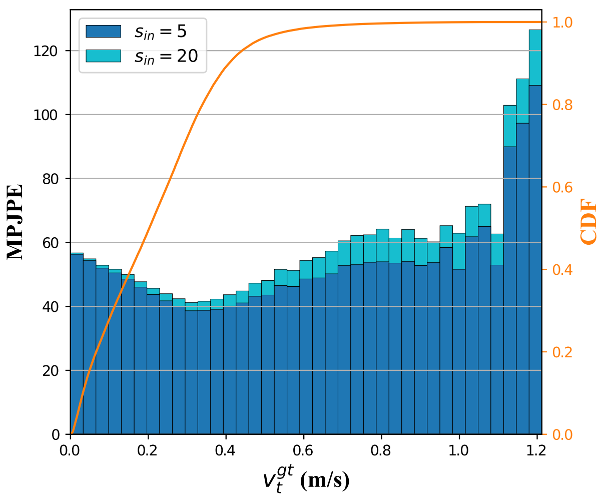

Figure 3: The MPJPE on Human3.6M in contrast to the ground truth pose velocity v_t^{gt}. The velocities are discretized into equally sized intervals. We additionally show the cumulative distribution (CDF) over the velocities in the dataset.

Table 4: Computational complexity and best MPJPE (mm) on Human3.6M with MoCap pre-training. 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.

N	s_{out}	s_{in}	FLOPs↓	PPS↑ (w/o CPN)	PPS↑ (w/ CPN)	MPJPE↓
81	2	4	564 M	326	105	44.8
		10	543 M	334	179	45.5
		20	**535 M**	337	234	47.9
351	5	5	1062 M	704	151	**42.6**
		10	999 M	759	255	43.1
		20	966 M	**827**	**399**	45.0

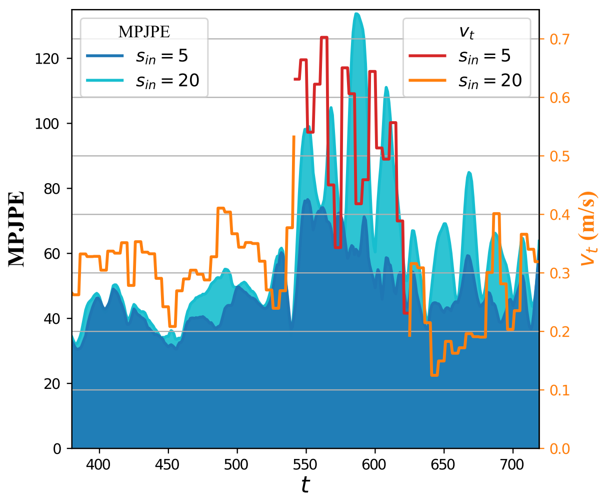

Figure 4: MPJPE on an exemplary "WalkDog" video from Human3.6M, with $N = 351$ and $s_{\text{out}} = 5$. We switch our model from $s_{\text{in}} = 20$ to $s_{\text{in}} = 5$ for video sections where the observed relative pose velocity v_t surpasses 0.5 m/s (red).

in daily life, it will not fit applications in *e.g.* sporting activities.

6. Adaptive Input Stride

Finally, we discuss further potential of our method that is yet to be exploited. One of the main advantages of our method is that a single instance of our uplifting model (*i.e.* a single set of model parameters) can support different input strides. Thus, a single model can be operated with different computational complexity and processing rates (see Table 4 for extended runtime and complexity results). For all experiments in this paper, the input stride is kept constant throughout the processing of an entire video. This is no strict requirement though. A change in input stride only affects how many video frames are used for 2D pose estimation and subsequent pose token generation within the spatial Transformer. No other reconfiguration of our model is required. Thus, the input stride can be changed online while processing a video stream. This enables hardware-limited devices to dynamically adapt to currently available shared resources like memory or computational units (CPUs, GPUs, TPUs).

A second use case of variable input strides is the adaption to the occurring human motion. Based on the discussion about movement speed in Section 5, we can process a video with long input stride by default, and only switch to a shorter input stride for increased precision when observing fast body movement. Figure 4 represents an exemplary Human3.6M video where a short sequence of running occurs. By thresholding the pose velocity v_t from the 3D pose estimates (orange), *e.g.* with 0.5 m/s, we can identify this video section and switch from $s_{\text{in}} = 20$ to $s_{\text{in}} = 5$. Only when the velocity (red) drops below the threshold for a fixed number of frames, we switch back to the more efficient input stride. This way we avoid the otherwise failed 3D pose estimation with an MPJPE > 80 mm. Note that the relative velocity is only one of many possible statistics for identifying difficult video sections. We leave the development and evaluation of such statistics as a research direction for future work.

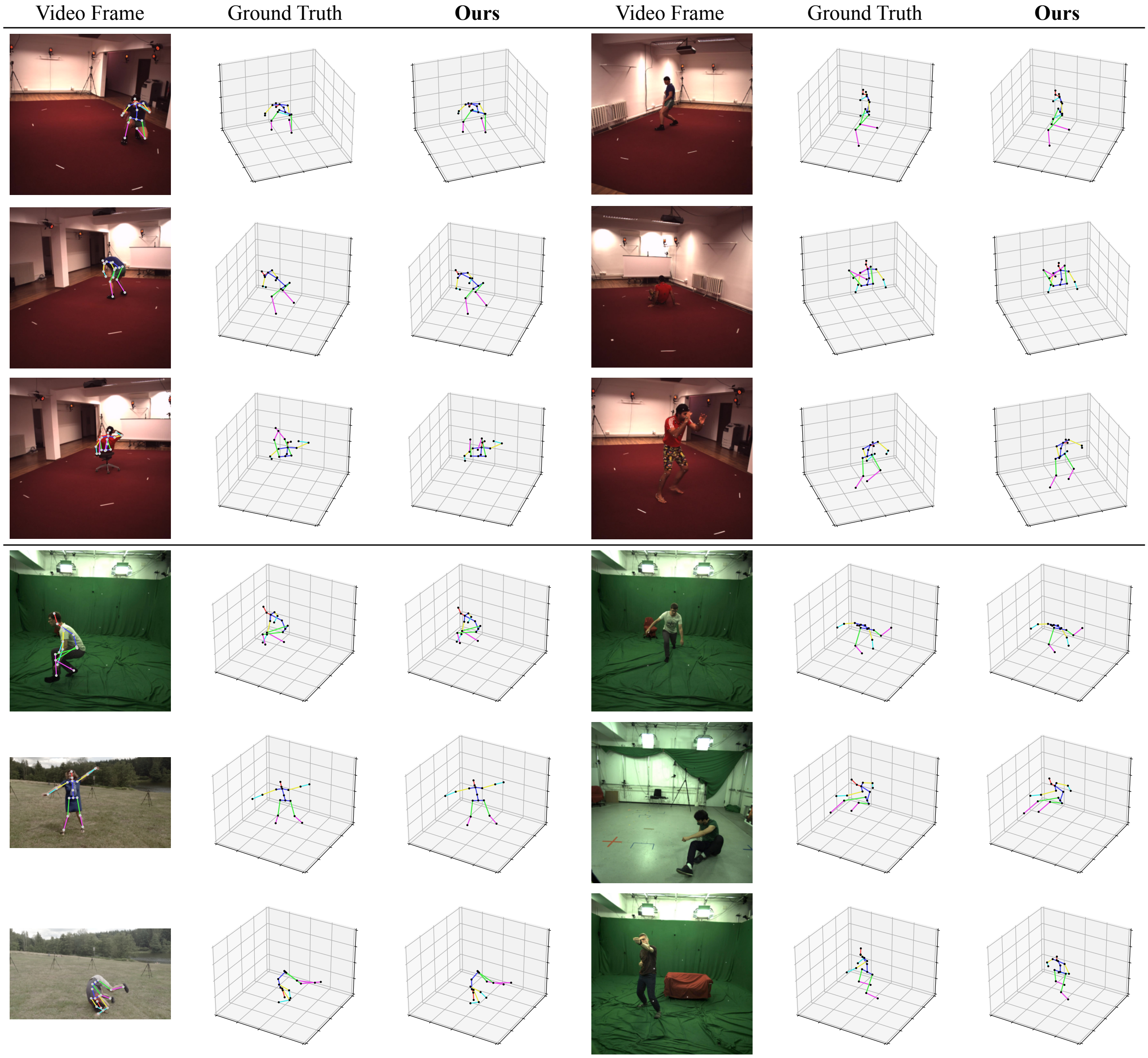

Figure 5: Qualitative examples on Human3.6M (top) and MPI-INF-3DHP [3] (bottom). The results are generated with our best models and 2D poses at 2.5 Hz. The left column shows results on key-frames, the right column on non-key-frames. Failure cases are depicted in rows three and six.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.