Realistic Full-Body Tracking from Sparse Observations via Joint-Level Modeling

Xiaozheng Zheng * Zhuo Su * Chao Wen Zhou Xue † Xiaojie Jin
ByteDance Inc

Abstract

To bridge the physical and virtual worlds for rapidly developed VR/AR applications, the ability to realistically drive 3D full-body avatars is of great significance. Although real-time body tracking with only the head-mounted displays (HMDs) and hand controllers is heavily underconstrained, a carefully designed end-to-end neural network is of great potential to solve the problem by learning from large-scale motion data. To this end, we propose a two-stage framework that can obtain accurate and smooth full-body motions with the three tracking signals of head and hands only. Our framework explicitly models the joint-level features in the first stage and utilizes them as spatiotemporal tokens for alternating spatial and temporal transformer blocks to capture joint-level correlations in the second stage. Furthermore, we design a set of loss terms to constrain the task of a high degree of freedom, such that we can exploit the potential of our joint-level modeling. With extensive experiments on the AMASS motion dataset and real-captured data, we validate the effectiveness of our designs and show our proposed method can achieve more accurate and smooth motion compared to existing approaches.

1. Introduction

Driving human avatars in VR/AR can help to bridge the gap between the physical and virtual worlds, and create a more natural and immersive user experience. However, in a typical capture setting, only the head and hands are tracked with Head Mounted Displays (HMD) and hand controllers. With limited inputs, driving the full-body avatar is inherently an underconstrained problem. Considerable endeavor has been dedicated to addressing the challenge of inferring full-body human pose exclusively through sparse AR/VR signals, head and hands. Although recent studies [5, 8, 9] have shown promising results, they are not suitable for real-time applications like VR body tracking. With real-time performance in mind, Winkler *et al*. [39] use reinforcement learning for training and outperform kinematic approaches with fewer artifacts. But their method requires future frames, which introduces latency to the system. Most recent work AvatarPoser [16] solves the problem in a more practical way by combining transformer-based architecture and inverse-kinetic optimization, setting the benchmark on large motion capture datasets (AMASS). Despite the success of AvatarPoser across a wide variety of motion classes, we argue that a learning-based end-to-end method provides more merits in simplicity, robustness, and generalization compared with a hybrid method.

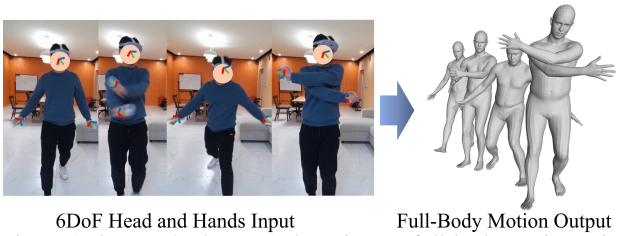

Figure 1. Our method accurately estimates full-body motion using only head and hand tracking signals.

Our key insight is that correlations between different body joints should be explicitly modeled for human pose estimation as body movements are highly structured and coordinated. Especially for the problem of estimating full-body motion from sparse observations, joint-level modeling is essential as the position and rotation of each joint can affect each other, and the overall body pose. By taking into account these correlations, we can derive more plausible full-body motion, even when observations are limited. Therefore, we design a two-stage joint-level modeling framework to capture these dependencies between body joints for more accurate and smoother human motion. In the first stage, we explicitly model the joint-level features. Then we utilize these features as spatiotemporal tokens in the second stage for a transformer-based network to capture the joint-level dependencies for recovering full-body motions.

In the first stage, we explicitly model the joint-level features as 1) *joint-rotation features* and 2) *joint-position features*. Joint-rotation offers higher compactness and computational efficiency, whereas joint-position enables more

Project page: https://zxz267.github.io/AvatarJLM.
*Equal contribution.
†Corresponding author.

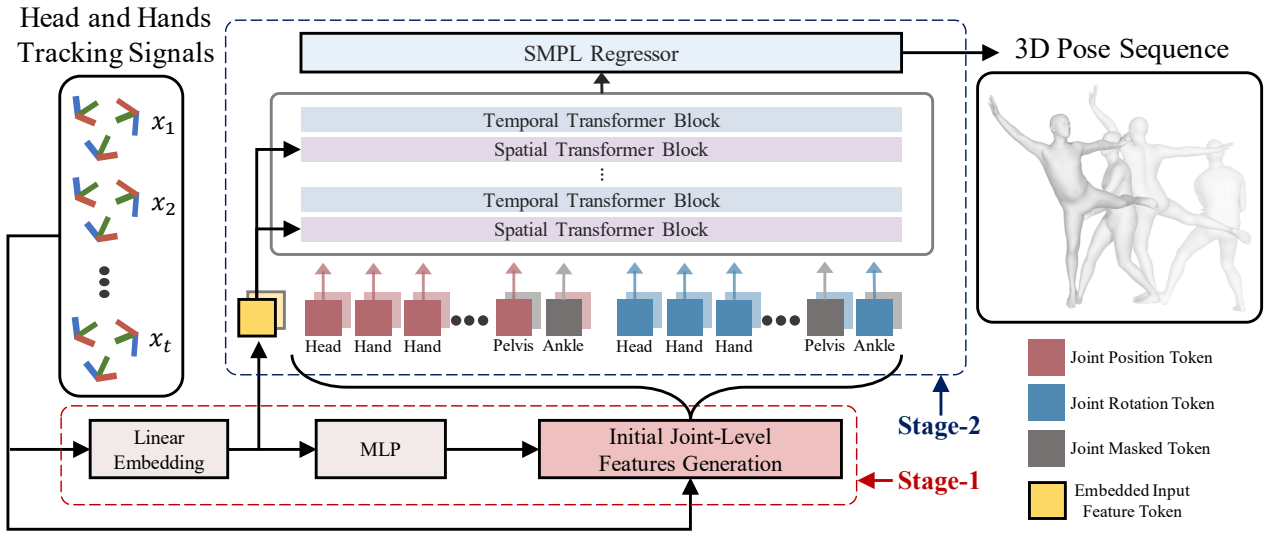

Figure 2. Illustration of our two-stage joint-level modeling framework. In the first stage, we embed a sequence of sparse signals to high-dimensional input features. Then, we utilize an MLP to obtain the initial full-body poses from the features. After that, we combine the initial full-body poses and the sparse input signals to generate initial joint-level features. In the second stage, we convert the initial joint-level features to joint-level tokens and then feed those tokens to a transformer-based network to capture the joint-level dependencies in spatial and temporal dimensions alternatively. In each spatial transformer block, we supplement an additional embedded input features token generated from the high-dimensional input features. Finally, we employ an SMPL regressor to transform the spatiotemporal modeled joint-level features into the 3D full-body pose sequence.

precise control and is intuitively easier to comprehend. By combining the advantages of both features, we attain a resilient and rational human motion with the improved ability to align endpoints more accurately.

In the second stage, we alternatively use spatial transformer blocks and temporal transformer blocks to capture the joint dependencies. Given the ambiguity inherent in our problem, we also opt to utilize the features from the first stage of each individual frame as an *embedded input features* (EIF) token for every single spatial transformer block to reinforce the influence of the input sparse observations.

To capitalize on the advantages of our joint-level modeling and mitigate the risk of overfitting in the highly under-constrained problem, we have incorporated a set of loss terms into our approach. These loss terms consist of hand alignment loss, motion smoothness loss, and physical loss, each meticulously designed to enhance the efficacy and generalizability of our body-tracking system in real-world scenarios, considering the intricate and uncertain problem nature.

Extensive experiments on the large-scale motion dataset AMASS [26] have demonstrated the effectiveness of our proposed designs. We also collect real-data samples for further qualitative and quantitative evaluations. Specifically, we conduct a thorough comparison of our approach against existing methods using various protocols. The comparative results show that our approach significantly outperforms existing methods in all protocols by a large margin. Moreover, our qualitative results demonstrate a significant improvement in accuracy and smoothness over the previous state-of-the-art approach, without the need for post-processing.

In summary, our contributions are the following:

- We propose a novel two-stage network that can effectively estimate full-body motion from the sparse head and hand tracking signals with high accuracy and temporal consistency. Note that our method does not need any post-processing and significantly outperforms existing state-of-the-art approaches.
- We elaborately design our feature extractor that generates joint-level rotational, positional, and embedded input features. These features are then utilized as spatiotemporal tokens and processed using a transformer-based network, which allows for better modeling of joint-level correlations.
- We introduce a set of losses that are tailored for the task of full-body motion estimation, and experimentally demonstrate the effectiveness of these losses in achieving high accuracy while avoiding undesirable artifacts such as floating, penetration, and skating.

2. Related Work

2.1. Full-Body Pose from Sparse Observations

The task of estimating the full-body pose of a human from sparse observations has garnered significant attention in the research community [7, 38, 15, 42, 41, 17, 2, 40, 9, 5, 16, 39]. Previous studies [38, 15, 42, 41, 17] have relied on

multiple IMU inputs to track the signals of the head, arms, pelvis and legs to capture the full body movements. SIP [38] demonstrates the possibility of reconstructing accurate 3D full-body poses using only 6 IMUs without any other information (e.g., video). Sequentially, DIP [15] further uses deep learning to achieve real-time performance and better accuracy with only 6 IMUs either. Most recently, there has been a shift towards a more AR/VR-focused setup, with some studies [2, 40, 9, 5, 16, 39] exploring the use of only HMD and hand controllers as input sources for even sparser observations.

Choutas *et al*. [8] propose a neural optimization method to fit a parametric human body model given the observations of the head and hands. Aliakbarian *et al*.'s recent work [5] uses the generative model, normalizing flow, to address the under-constrained problem. Dittadi *et al*. [9] take advantage of the Variational Autoencoders to synthesize the lower body and implicitly assume the fourth 3-D input by encoding all joints relative to the pelvis.

The work QuestSim [39] and AvatarPoser [16] bear the most similarities to our own methodology. QuestSim employs reinforcement learning to facilitate physics simulation. Nonetheless, the approach requires additional future information at runtime to achieve optimal results. AvatarPoser uses a simple transformer-based network architecture to achieve accurate and smooth full-body motions, which demonstrates the feasibility of using a discriminative method for solving the task of full-body pose estimation from sparse observations.

Following the work by Jiang *et al*. [16], we further investigate the potential of discriminative methods in full-body pose estimation from sparse observations. Our approach takes into account the nature of the task in the design of the framework, allowing us to leverage discriminative models for achieving more accurate and smooth full-body motions from only head and two-hand tracking signals.

2.2. Transformer for Human Pose Estimation

Transformer [37] has emerged as a popular tool for human pose estimation tasks in recent years. Specifically, a number of studies [14, 21, 22, 25] have utilized transformers to solve the task of 3D human pose estimation from RGB images. In addition, several studies [45, 44, 20] have employed transformers to lift 2D human pose to the corresponding 3D human pose. Transformers have also been widely used in motion generation tasks, as seen in studies [4, 34, 30].

Two works [16, 17] have utilized transformer-based methods to address a problem similar to that which our proposition seeks to solve. The work from Jiang *et al*. [17] employs a transformer network and a recurrent neural network in combination to accurately estimate full-body motion using data from six IMUs. Meanwhile, Jiang *et al*. [16] solve the pose estimation from HMD with a combination of a transformer network followed by traditional model-based optimization. However, both of them only treat each frame of the observed signals as a token for the temporal transformer to process. This way does not take the task nature into account and does not model the joint-level features, which makes them hard to obtain more accurate and smooth full-body motions.

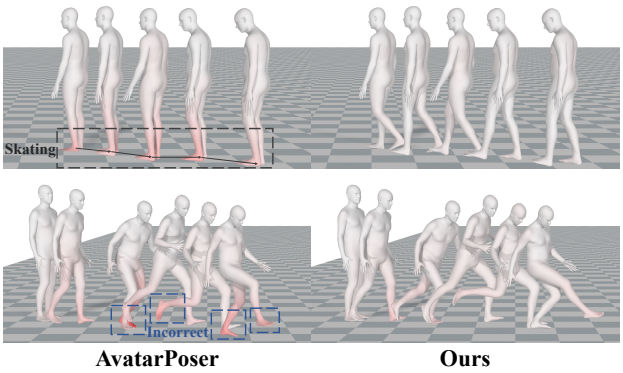

Figure 3. Qualitative comparisons between AvatarPoser and ours. The results are color-coded to show errors in red.

Both of these methods process observed signals as sequences of temporal tokens, which are then utilized as inputs to the transformer network in a direct manner. We argue that the direct application of input signals renders the under-constrained problem more difficult to solve. In response, we explicitly model the joint-rotation and joint-position features to enable a more robust, intuitive, and comprehensive representation. Furthermore, we utilize a transformer-based network that alternatively captures spatial and temporal joint dependencies, in a manner similar to the approach taken by MixSTE [44].

3. Method

3.1. Problem Formulation

In this task, we aim at predicting the full-body motion $\boldsymbol{\Theta} = \{\theta_i\}_{i=1}^{t} \in \mathbb{R}^{t\times s}$, given a sequence of sparse tracking signals $\boldsymbol{X} = \{x_i\}_{i=1}^{t} \in \mathbb{R}^{t\times c}$ of t frames from headsets and hand controllers, where c and s denote the dimension of the input and output. Following the previous work [16], we take the position, rotation, positional velocity, and angular velocity of the head and hands as the input signals; adopt pose parameters of the first 22 joints of the SMPL [24] model to represent the output; and use the 6D representation of rotations for the input and SMPL model for its effectiveness [46]. Therefore, the input and output dimensions are $c = 3\times(6+6+3+3) = 54$ and $s = 22\times 6 = 132$.

3.2. Two-Stage Joint-Level Modeling Framework

As illustrated in Fig. 2, our joint-level modeling consists of two stages: 1) the initial joint-level features generation stage and 2) the joint-level spatiotemporal correlation mod-

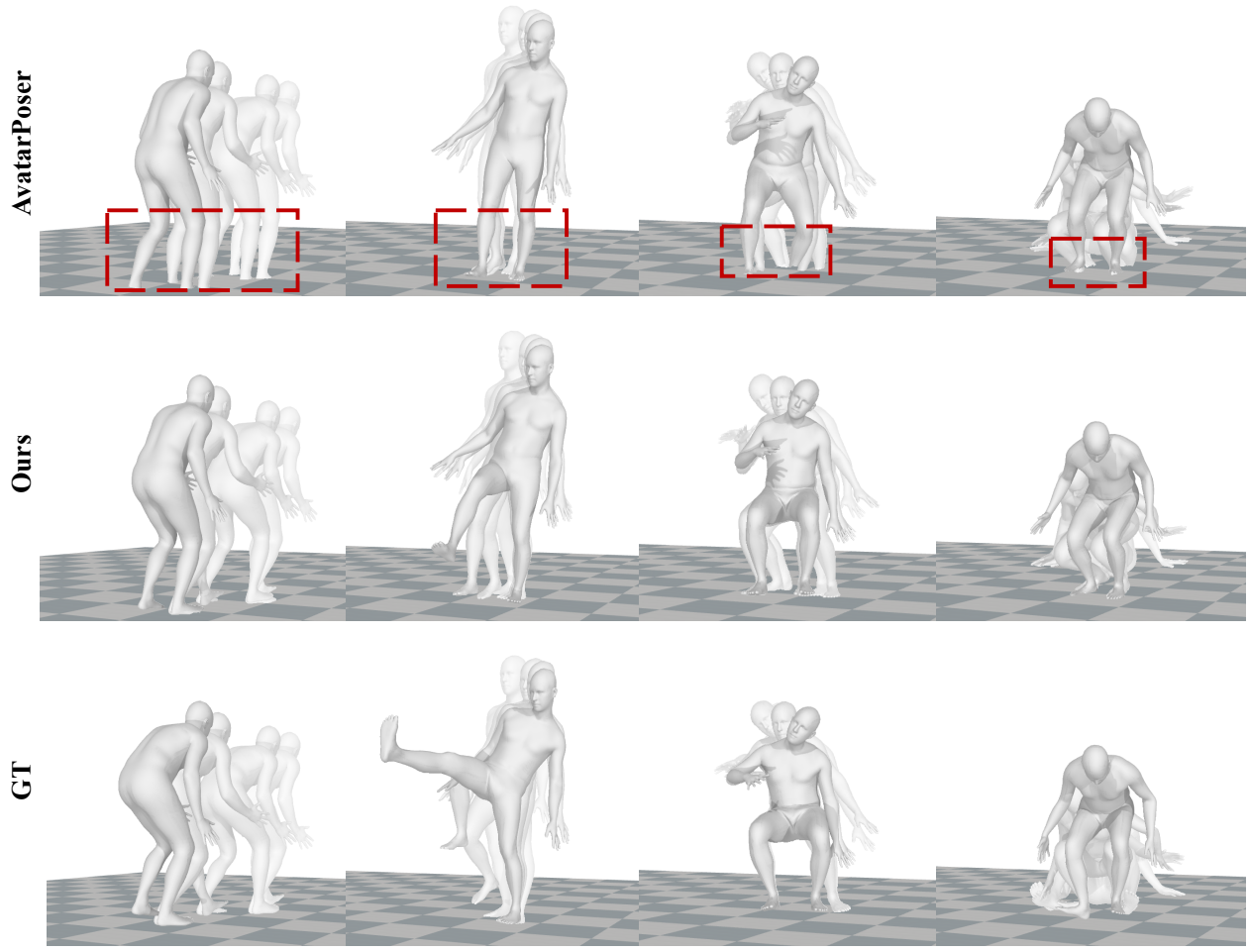

Figure 4. Qualitative comparisons between AvataPoser, our method, and ground-truth on real-captured data.

eling stage. The first stage uses a simple network upon the input signals to generate the initial joint-level features serving as a basis for further exploiting the joint correlation. To achieve spatially accurate and temporally consistent full-body human motion, the second stage captures the joint dependencies in both spatial and temporal dimensions based on a spatiotemporal transformer-based network.

3.2.1 Initial Joint-Level Features Generation

To enable the explicit modeling of joint-level correlations, it is essential to generate initial joint-level features. We accomplish this by 1) generating initial full-body poses and then 2) transforming them into joint-level positional and rotational features. In this way, these two sorts of representations can explore different feature spaces to benefit following spatiotemporal modeling.

Specifically, the initial full-body pose for each frame is generated using two MLPs as follows:

$$\begin{aligned} \boldsymbol{H}_{embed} &= F_{embed}(\boldsymbol{X}) \in \mathbb{R}^{t\times d_1}, \\ \boldsymbol{\Theta}_{init} &= F_{reg}(\boldsymbol{H}_{embed}) \in \mathbb{R}^{t\times 22\times 6}, \end{aligned} \tag{1}$$

in which F_{embed} embeds the input signals to high-dimensional input features $\boldsymbol{H}_{embed}$, F_{reg} further regress the initial full-body pose $\boldsymbol{\Theta}_{init}$, and d_1 represents the dimension of the high-dimensional input features.

Next, we perform a forward-kinematics on the initial poses $\boldsymbol{\Theta}_{init}$ to obtain 3D positions of the joints $\boldsymbol{P}_{init} \in \mathbb{R}^{t\times 22\times 3}$. We subsequently convert these positions from the SMPL coordinate system to the head-relative coordinate system and convert the relative joint rotations to global joint rotations. Since we have specific observations of the head and hands, we substitute the positions and rotations of these joints with tracking signals, which provide guidance for the subsequent capture of joint correlations. Then, we utilize two linear layers to embed the joint rotations and positions into high-dimensional joint-level features denoted as $\boldsymbol{H}_{\Theta} \in \mathbb{R}^{t\times 22\times d_2}$ and $\boldsymbol{H}_{P} \in \mathbb{R}^{t\times 22\times d_2}$, respectively. Here, d_2 is the dimension of the high-dimensional joint-level features. Finally, we concatenate $\boldsymbol{H}_{\Theta}$ and $\boldsymbol{H}_{P}$ to obtain the initial joint-level features $\boldsymbol{H}_{init} = [\boldsymbol{H}_{\Theta} \otimes \boldsymbol{H}_{P}] \in \mathbb{R}^{t\times 44\times d_2}$.

3.2.2 Spatiotemporal Correlations Modeling

To achieve spatially accurate and temporally consistent full-body human motion, we fully exploit the joint-level features in both spatial and temporal dimensions to model the spatiotemporal correlations. Observed that our targeting full-body motion is dependent on the input signals from the head and hands, the most potential joint dependencies are related to the correlations between different joints and the head and hand joints. To fully capture these correlations, we adopt a transformer-based network allowing easy long-range dependencies modeling between other joints and the head and hands joints. Therefore, the initial joint-level features are then utilized as carefully designed tokens for the spatiotemporal transformer-based network.

To fully excavate implicit information of designed tokens, we introduce a Spatial Transformer Block (STB) F_{stb} and a Temporal Transformer Block (TTB) F_{ttb} to capture the dependencies in the spatial and temporal dimensions, respectively. STB aims at capturing the spatial joint dependencies within a frame, especially the dependencies between other joints and observed head and hands joints, and achieving a reasonable single-frame pose estimation. The input joint-level tokens $h_i \in \mathbb{R}^{44\times d_2}$ include the joint rotational/positional features within a frame, where h_i denotes the initial joint-level features $\boldsymbol{H}_{init}$ in the i^{th} frame. To preserve certain input information during the feature learning process, we supplement additional *embedded input features* token f_i to every single STB, where f_i is the i^{th} frame of $\boldsymbol{H}_f \in \mathbb{R}^{t\times 1\times d_2}$ linear-transformed from $\boldsymbol{H}_{embed}$. Therefore, the input tokens for STB are $s_i = [h_i \otimes f_i] \in \mathbb{R}^{45\times d_2}$ Note that before feeding s_i into STB, we add a learnable spatial positional encoding $e_s \in \mathbb{R}^{45\times d_2}$ to s_i for indicating the relative location of the joint rotations/positions. Then, STB utilizes the self-attention mechanism [37] to model the dependencies of all the tokens for each frame and outputs the spatially enhanced joint-level features $\boldsymbol{H}_s = \{F_{stb}(s_i)^{1:44}\}_{i=1}^{t} \in \mathbb{R}^{t\times 44\times d_2}$.

TTB focuses on learning the temporal correlations of each joint for maintaining temporal consistency and motion accuracy. TTB treats each kind of feature across the sequence as tokens, resulting in the features $h^i \in \mathbb{R}^{t\times d_2}$ with t tokens in total, where h^i denotes the initial joint-level features $\boldsymbol{H}_{init}$ sliced in the second dimension. Besides, we add a learnable temporal positional encoding $e_t \in \mathbb{R}^{t\times d_2}$ to $h^i \in \mathbb{R}^{t\times d_2}$ for indicating the location of a specific joint feature in the sequence. Then the temporally enhanced joint-level features $\boldsymbol{H_t} = \{F_{ttb}(h^i)\}_{i=1}^{44} \in \mathbb{R}^{t\times 44\times d_2}$ are encoded by the TTB as outputs.

Spatial and temporal correlations modeling complement each other, in which STB tends to generate reasonable pose without temporal consistency and TTB tends to smooth the motion while introducing pose misalignment. Inspired by MixSTE [44], we alternatively use STB and TTB, which can decompose the feature learning into spatial and temporal dimensions. Specifically, we stack STB and TTB for n loops to obtain the final spatiotemporally modeled features $\boldsymbol{H_{st}} \in \mathbb{R}^{t\times 44\times d_2}$.

Finally, we employ an MLP to regress the pose parameters $\boldsymbol{\Theta}$ of SMPL from $\boldsymbol{H_{st}}$. The MLP consists of 2 linear layers, 1 group normalization layer, and 1 activation layer. The local full-body pose $\boldsymbol{P}$ is derived from $\boldsymbol{\Theta}$ using SMPL.

Input	Method	MPJRE	MPJPE	MPJVE
Four	Final IK	12.39	9.54	36.73
	CoolMoves	4.58	5.55	65.28
	LoBSTr	8.09	5.56	30.12
	VAE-HMD	3.12	3.51	28.23
	AvatarPoser	2.59	2.61	22.16
	Ours	**2.40**	**2.09**	**17.82**
Three	Final IK	16.77	18.09	59.24
	CoolMoves	5.20	7.83	100.54
	LoBSTr	10.69	9.02	44.97
	VAE-HMD	4.11	6.83	37.99
	AvatarPoser	3.21	4.18	29.40
	Ours	**2.90**	**3.35**	**20.79**

Table 1. Evaluation results under Protocol 1.

Dataset	Method	MPJRE	MPJPE	MPJVE
CMU	Final IK	17.80	18.82	56.83
	CoolMoves	9.20	18.77	139.17
	LoBSTr	12.51	12.96	49.94
	VAE-HMD	6.53	13.04	51.69
	AvatarPoser	5.93	8.37	35.76
	Ours	**5.34**	**7.75**	**26.54**
BMLrub	Final IK	15.93	17.58	60.64
	CoolMoves	7.93	13.30	134.77
	LoBSTr	10.79	11.00	60.74
	VAE-HMD	5.34	9.69	51.80
	AvatarPoser	4.92	7.04	43.70
	Ours	**4.71**	**6.49**	**36.96**
HDM05	Final IK	18.64	18.43	62.39
	CoolMoves	9.47	17.90	140.61
	LoBSTr	13.17	11.94	48.26
	VAE-HMD	6.45	10.21	40.07
	AvatarPoser	6.39	8.05	30.85
	Ours	**5.86**	**6.60**	**23.57**

Table 2. Evaluation results under Protocol 2.

3.3. Loss Design and Training Process

We make use of $L1$ body orientation loss, $L1$ body joint rotational loss, and $L1$ body joint positional loss, resembling AvatarPoser [16]. Moreover, since this task is of a high degree of freedom, it is not easy to obtain accurate motion without additional constraints. Therefore, to better exploit the potential of our joint-level modeling, we introduce

Method	MPJRE	MPJPE	MPJVE	Jitter
VPoser-HMD†	/	6.74	/	/
HuMoR-HMD †	/	5.50	/	/
VAE-HMD†	/	7.45	/	/
ProHMR-HMD†	/	5.22	/	/
FLAG†	/	4.96	/	/
AvatarPoser*	4.56	6.44	34.45	11.15
Ours	**4.30**	**4.93**	**26.17**	**7.19**

Table 3. Evaluation results under Protocol 3. † denotes methods explicitly using additional pelvis information during inference. * denotes our retrained AvatarPoser using their public source code.

a set of losses that are tailored for this task to achieve better alignment with the input signals, temporal consistency, and physically plausible results.

Hand Alignment Loss. As discussed in the previous study [16], predicting absolute pelvis translations is worse than obtaining the translation from the known head position. Therefore, we also obtain the translation from *head alignment* that aligns the head position of our local full-body pose to the global position of the head to obtain full-body positions in the global coordinate system $\boldsymbol{P}^g$. However, this approach causes a misalignment between the predicted and input global hand positions. To solve this problem, AvatarPoser adopts an inverse kinematics (IK) module to align hands. Nevertheless, the IK module is very slow. To better address this issue, we add a hand alignment loss $L_h = \frac{1}{2}\sum_{i=1}^{t}\|p_i^{hand} - \hat{p}_i^{hand}\|_1$ to align the global hands after the head alignment, where p_i^{hand} denotes global hand joint positions of i^{th} frame from $\boldsymbol{P}^g$ and $\hat{}$ denotes ground-truth. This way makes the whole framework end-to-end trainable, contributing to good hand alignment without using the IK module to slow down the system. Since the full-body pose is highly correlated to the hand joints, our performance in different metrics all obtain improvement when the accurate hand position is perceived by the network.

Motion Loss. Following previous studies [42, 34], we also utilize velocity loss $L_v(l) = \frac{1}{t-1}\sum_{i=1}^{t-1}\|(p_{i+l} - p_i) - (\hat{p}_{i+l} - \hat{p}_i)\|_1$ and foot contact loss $L_{fc} = \frac{1}{t-1}\sum_{i=1}^{t-1}\|(p_{i+1}^{feet} - p_i^{feet}) \cdot m_i\|_1$ for achieving a smooth and accurate motion, where p_i is the predicted global full-body pose in i^{th} frame and p_i^{feet} denotes the joints relevant to the feet. L_v encourages the inter-frame velocity to be close to the corresponding velocity of the ground-truth; L_{fc} enforces zero feet velocity when the feet are on the ground. $m_i \in \{0,1\}^k$ is the binary foot contact mask of i^{th} frame, denoting whether the feet touch the ground and k is the feet-relevant joint number. Instead of only using $L_v(1)$ to supervise the velocity between adjacent frames, we also utilize $L_v(3)$ and $L_v(5)$ to avoid the accumulated velocity error. Therefore, the proposed motion loss is defined as :

$$L_{mot} = L_v(1) + L_v(3) + L_v(5) + L_{fc}. \tag{2}$$

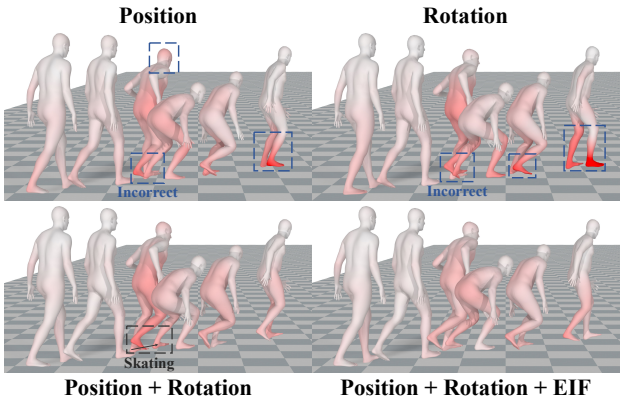

Figure 5. Ablation study for our method with four different generated features for the second stage, in which the errors are color-coded in red.

Physical Loss. Since this task lacks lower body information, the prediction tends to fail to match the upper body while maintaining the lower body physically plausible for challenging cases (e.g., jump, sit). To mitigate physical implausibility (especially ground penetration), we add ground penetration loss $L_p = \frac{1}{t}\sum_{i=1}^{t}\|(z_{p_i}^{min} - z_{ground}) \cdot l_i\|_1$ and ground-foot height loss $L_{fh} = \frac{1}{t}\sum_{i=1}^{t}\|(z_{p_i}^{feet} - z_{ground}) \cdot m_i\|_1$, where z_{ground} is the ground height, $z_{p_i}^{min}$ is the predicted height of the lowest joint in i_{th} frame, $z_{p_i}^{feet}$ is the predicted height of the feet-relevant joints in i^{th} frame, and $l_i \in \{0,1\}^{22}$ denotes whether the joint is lower than the ground. Thus, the proposed physical loss is defined as:

$$L_{phy} = L_p + \alpha L_{fh}. \tag{3}$$

These two physical losses are complementary to each other because penetration error tends to "push" the predictions away from the ground, while the foot height error tends to "pull" the predictions to the ground.

Overall Loss. Our complete loss function for training the model is defined as follows:

$$L = L_{first} + \beta L_{ori} + \gamma L_{rot} + \delta L_{pos} + \epsilon L_h + \zeta L_{mot} + L_{phy}, \tag{4}$$

where L_{first}, L_{ori}, L_{rot}, and L_{pos} are $L1$ loss for the initial head-relative full-body pose, the final SMPL root orientation $\boldsymbol{\Theta}^{1:6}$, joint rotations $\boldsymbol{\Theta}^{7:132}$, and full-body pose $\boldsymbol{P}$. We set α, β, γ, δ, ϵ, and ζ to 0.5, 0.02, 2, 5, 5, and 50 respectively to balance their loss scale.

Masked Training. During training, we randomly masked 2 of the tokens except for the head and hands tokens. In this way, the model is more robust for the quality of the generated initial joint-level features.

Implementation Detail. Following AvatarPoser [16], we set the input sequence length t to 41 frames if not stated otherwise. During inference, we utilize sliding windows for prediction like AvatarPoser, which predicts the current

Protocol	Method	MPJRE	MPJPE	MPJVE	Jitter	Ground	Skate	H-PE	U-PE	L-PE
1	AvatarPoser*	3.07	4.15	28.39	16.15	3.80	0.23	2.45	2.00	7.91
	Ours	**2.90**	**3.35**	**20.79**	**8.39**	**3.30**	**0.13**	**1.24**	**1.72**	**6.20**
3	AvatarPoser*	4.56	6.44	34.45	11.15	2.95	0.32	3.70	2.93	12.59
	Ours	**4.30**	**4.93**	**26.17**	**7.19**	**2.17**	**0.21**	**1.45**	**2.27**	**9.59**

Table 4. More metrics comparisons with AvatarPoser [16] under Protocol 1 and Protocol 3. * denotes our retrained AvatarPoser using their public source code.

Method	MPJRE	MPJPE	MPJVE	Jitter	Ground	Skate	H-PE	U-PE	L-PE
AvatarPoser*	7.28	11.22	31.67	12.87	2.20	0.30	6.60	5.79	20.73
Ours	**6.98**	**9.52**	**25.78**	**10.04**	**0.20**	**0.21**	**5.31**	**5.16**	**17.15**

Table 5. Evaluation results on the real-captured data. * denotes our retrained AvatarPoser using their public source code.

frame using the current frame and previous $t-1$ frames. We train the network with a batch size of 64 and use Adam solver [18] for optimization. Our model is trained for $100,000$ iterations. The learning rate starts at $1e-4$ and drops to $1e-5$ after $60,000$ iterations. The feature dimension d_1 and d_2 for our model are set to 1024 and 512, and the stacked loops n of our network are set to 6. Our model takes $24.7ms$ to infer 41 frames on an NVIDIA A100 GPU without needing any further post-processing (e.g., IK).

4. Experiment

In this section, we first report the metrics adopted in previous tasks [16, 5, 43, 42]. Then we compare with previous state-of-the-art methods qualitatively and quantitatively. We also evaluate each of our main contributions. Fig. 4 demonstrates the superiority of our methods using various motions. Please kindly refer to our supplementary video/materials for more details.

Accuracy-related metric. To evaluate the pose accuracy for each frame, we use *MPJPE* (mean per joint position error) and *MPJRE* (mean per joint rotation error), which measure the average position/relative rotation error of all body joints. Besides the full-body MPJPE, we also evaluate upper-body MPJPE (*U-PE*), lower-body MPJPE (*L-PE*), and hand MPJPE (*H-PE*), which can reflect different capabilities of the methods.

Smoothness-related metric. To evaluate the motion smoothness, we use *MPJVE* (mean per joint velocity error) and *Jitter*. *MPJVE* measures the average velocity error of all body joints. *Jitter* measures the average jerk (time derivative of acceleration) of all body joints, which reflects the smoothness of the motion [11].

Physics-related metric. To evaluate the physical plausibility of motions, we use *Ground* and *Skate* metrics. *Ground* measures the distance between the lowest ground-truth body joint and the lowest predicted body joint. This metric reflects whether the generated motions have the correct contacting relationship with the ground (if the body penetrates the ground or floats above the ground, the error will be large). *Skate* measures the average horizontal displacement between the grounding feet in adjacent frames.

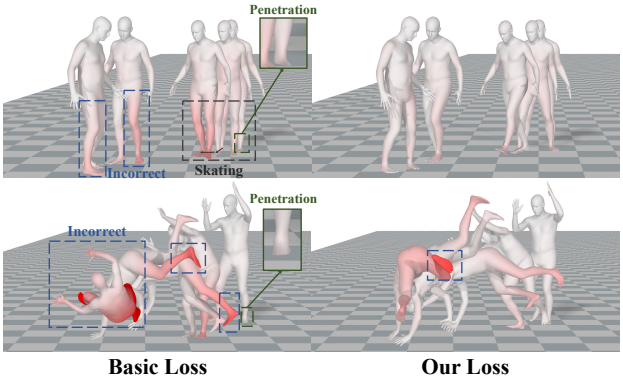

Figure 6. Ablation study for our method with different loss settings, in which the errors are color-coded in red.

4.1. Comparison

We make a thorough comparison with previous studies on the AMASS dataset[26]. Fig. 3 presents the qualitative comparison against AvatarPoser [16] and demonstrates that our method can achieve more accurate and smooth results without physical implausibilities. For quantitative comparison, we adopt three widely used settings as follows.

Protocol 1. For the first setting, we follow [16] to split the subsets CMU [13], BMLrub [35], and HDM05 [28] into 90% training data and 10% testing data. We compare our performance using both three (headset and controllers) and four inputs (add a pelvis tracker) with previous methods [16, 9, 2, 40, 32]. Note that unless otherwise stated, we all use three inputs for experiments. As shown in Tab. 1, our performance with both three and four inputs outperforms existing approaches by a large margin in all previously used metrics. At the top of Tab. 4, we present comprehensive comparisons between our method and AvatarPoser using all the metrics. The results show that our method can accurately estimate lower body motions (*L-PE*) while being much smoother and more physically plausible.

Protocol 2. For the second setting, we follow [16] to

Stage	Method	MPJRE	MPJPE	MPJVE	Jitter	Ground	Skate	H-PE	U-PE	L-PE
1	Constant Token	6.97	9.00	31.21	3.80	12.02	0.36	1.55	3.71	18.26
	Learnable Token	6.91	8.91	30.82	3.48	11.52	0.36	1.57	3.71	18.01
2	Position	6.15	6.77	25.53	5.76	2.63	0.23	**1.62**	**3.48**	12.51
	Rotation	6.00	7.48	26.41	**3.77**	2.48	0.26	1.92	3.62	14.21
	Rotation + Position	5.90	6.71	23.97	4.42	2.60	0.22	1.71	3.51	12.30
	Rotation + Position + EIF	**5.86**	**6.60**	**23.57**	4.10	**2.46**	**0.21**	1.69	3.52	**12.12**

Table 6. Performance comparisons between our proposed method with different initial joint-level features. The best results are in **bold**, and the second-best results are underlined.

Method	MPJRE	MPJPE	MPJVE	Jitter	Ground	Skate	H-PE	U-PE	L-PE
Ours + Basic Loss	6.09	7.50	32.53	8.98	3.66	0.35	3.11	3.93	13.76
+ Hand	5.87	6.90	29.07	6.88	3.73	0.33	**1.58**	**3.51**	12.85
+ Hand + Motion	**5.81**	6.74	24.25	4.22	3.22	0.22	1.60	3.56	12.32
+ Hand + Motion + Physical	5.86	**6.60**	**23.57**	**4.10**	**2.46**	**0.21**	1.69	3.52	**12.12**

Table 7. Performance comparisons between our proposed method with different loss functions. * denotes our retrained AvatarPoser using their public source code. The best results are in **bold**, and the second-best results are underlined.

perform a 3-fold cross-dataset evaluation to compare with [16, 9, 2, 40, 32]. Using the subsets CMU [13], BMLrub [35], and HDM05 [28], we train on two subsets and test on the other subset in a round-robin fashion. Tab. 2 shows the experimental results. Our method achieves the best performance over all the previously used metrics in all three datasets. Our performance is significantly better than the second-best method, demonstrating exploiting joint-level features contributes to generalization ability greatly.

Protocol 3. For the third setting, we follow [5] to use the subsets [13, 3, 36, 10, 27, 35, 27, 12, 26, 23, 1, 28] for training, and use the Transition [26] and HumanEva [33] subsets for testing. The comparative results with [29, 31, 19, 9, 16] are shown in Tab. 3. Since some existing approaches [29, 31, 19, 9] implicitly assume the knowledge of the pelvis position but we do not, the comparisons are unfair. However, our *MPJPE* is still superior to FLAG [5], showing the effectiveness of our method. When fairly compared with AvatarPoser, our method is significantly better. The performance gap between AvatarPoser and ours is larger than that in other protocols, which indicates our method can benefit from more training data. At the bottom of Tab. 4, we present the comprehensive comparisons between our method and AvatarPoser using all the metrics. Similar to Protocol 1, our proposed method outperforms AvatarPoser in all metrics by a large margin.

Moreover, to further evaluate our performance on the real headset-and-controllers data for VR/AR applications. We also capture a set of real evaluation samples from HMD and controller devices with the corresponding ground truth using a synchronized MoCap system; for details see supplementary materials. To quantitatively compare our method with AvatarPoser and show the sim-to-real performance gap on real-captured data, we use the models trained with Protocol 3. As shown in Tab. 5, even though there are some acceptable drops in certain metrics from synthetic data, our model still outperforms AvatarPoser by a large margin as well, indicating our performance improvement does not come from simply overfitting the synthetic data but from well-learned motion knowledge from the large-scale motion data. Fig. 4 qualitatively demonstrates that our model can reconstruct more realistic and physically plausible results for those challenging cases (e.g., walking backward, kicking, sitting, and standing up) better than AvatarPoser.

Beyond the above comparisons, we also conducted a user study to compare the subjective quality of our method with AvatarPoser. Our method achieved 3.69 scores while AvatarPoser gained 1.98 scores only (5-level Likert scale). More details are in supplementary materials.

4.2. Ablation Study

We perform ablation studies using CMU [13] and BMLrub [35] for training and HDM05 [28] for testing.

Initial joint-level features and tokens. To validate the effectiveness of our two-stage framework using a coarse full-body for feature initialization apart from the head and hand joints, we compare our method with two alternatives: 1) using the constant token as the initialized features; 2) using the learnable token as the initialized features, similar to [6, 25]. As shown in Tab. 6, these two alternatives are much worse than ours. Next, we demonstrate the effectiveness of our token design with different initial joint-level features. The results in Tab. 6 proves that using either *joint-rotation features* or *joint-position features* is worse than combining them together to exploit different useful features. After adding *embedded input features* token to provide input information for the spatial transformer blocks, the performance is better, indicating that repeatedly introducing the input information is useful for this task. Fig. 5 shows visual results with different initial joint-level features.

Method	MPJRE	MPJPE	MPJVE	Jitter	Ground	Skate	H-PE	U-PE	L-PE
AvatarPoser	6.39	8.05	30.85	-	-	-	-	-	-
AvatarPoser-L*	5.95	7.80	30.82	6.89	3.83	0.32	4.29	4.19	14.12
AvatarPoser-L* + Our Loss	6.02	7.14	23.92	**3.41**	2.51	**0.21**	1.82	3.54	13.43
Ours + Our Loss	**5.86**	**6.60**	**23.57**	4.10	**2.46**	**0.21**	**1.69**	**3.52**	**12.12**

Table 8. Architecture comparisons with AvatarPoser [16]. * denotes our retrained AvatarPoser using their public source code. AvatarPoser-L denotes a larger version of AvatarPoser.

Method	MPJRE	MPJPE	MPJVE	Jitter
Ours	**5.86**	**6.60**	**23.57**	**4.10**
w/o Mask Training	5.91	6.65	23.87	4.19

Table 9. Performance comparisons between our proposed method with and without mask training.

Loss. Tab. 7 shows the contribution of our designed loss functions. Adding hand alignment loss upon the basic loss significantly reduces hand error (*H-PE*) and makes the network end-to-end trainable, which also improves other metrics. Additional motion loss contributes a lot to the motion-related metrics (MPJVE, Jitter, and Sliding), and also benefit other metrics. Using physical loss, physics-related errors decrease, especially the *Ground* error. For more experiments on every single combination of all our loss functions, please refer to our supplementary materials. Fig. 6 provides our visual results with and without our designed loss. To better justify the effectiveness of our loss design, we also apply our complete loss function to the AvatarPoser. As shown in Tab. 8, our loss can also dramatically improve AvatarPoser, indicating that our loss design is not only useful for our network design but also suitable for this task.

Architecture. To fully compare our joint-level architecture with AvatarPoser, we enlarge AvatarPoser by using 12 transformer layers and setting the feature dimension to 512 for it (if even larger, AvatarPoser cannot converge). As shown in Tab. 8, our method still outperforms AvatarPoser to a great extent. Even after being significantly improved by our loss function, AvatarPoser still achieves an inferior performance to our method in almost all the metrics. Even though the Jitter metric for AvatarPoser is better, analyzing this metric alone makes no sense since Jitter error can be quite low when predicting the same wrong full-body poses. This phenomenon further justifies our joint-level modeling.

Mask training. Tab. 9 shows that our method with and without mask training. The results demonstrate that our method benefits from mask training. Meanwhile, this approach does not introduce any inference burden.

4.3. Limitation and Discussion

We expect to generate realistic motions, in which "realistic" emphasizes smoothness, physical plausibility, and alignment with the inputs. Despite our ability to achieve this objective in the majority of cases, there remain some issues that are beyond our capacity to solve. Fig. 7 presents our failure cases. On AMASS, our model can not predict complex lower body motions (e.g., ballet) and may collapse with rare motions (e.g., falling down). On real-captured data, our model has difficulty reconstructing motions with small variances of tracking signals but large lower-body movements. Inferring full-body poses from three sparse signals is under-constrained and highly dependent on large-scale motion data. Incorporating additional legs' signals (e.g., IMUs and images) may help resolve this problem. Furthermore, we believe more real-captured training data can also contribute greatly to the task since we observe some challenging cases can be solved with enough trained data on the synthetic data.

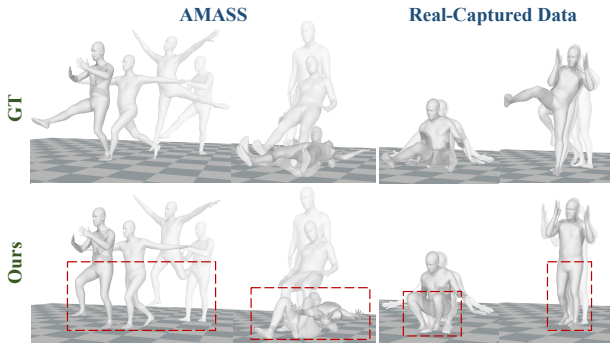

Figure 7. Failure cases on AMASS and real-captured data.

5. Conclusion

In this paper, we propose a two-stage learning-based framework for accurately estimating full-body motion from sparse head and hand tracking signals. We explicitly model the joint-level features in the first stage and then utilize them as spatiotemporal tokens for alternating spatial and temporal transformer blocks to estimate the full-body motion in the second stage. With a set of carefully designed losses, we fully exploit the potential of our joint-level modeling to obtain realistic full-body motion. Extensive experiments demonstrate both significant quantitative and qualitative improvement of our method over the previous state-of-the-art approaches without any post-processing during inference. We believe that our approach is a critical step in bridging the physical and virtual worlds for VR/AR applications.

References

[1] Sfu motion capture database. http://mocap.cs.sfu.ca. 8

[2] K. Ahuja, E. Ofek, M. Gonzalez-Franco, C. Holz, and A. D. Wilson. Coolmoves: User motion accentuation in virtual reality. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(2):1–23, 2021. 2, 3, 7, 8

[3] I. Akhter and M. J. Black. Pose-conditioned joint angle limits for 3d human pose reconstruction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1446–1455, 2015. 8

[4] E. Aksan, M. Kaufmann, P. Cao, and O. Hilliges. A spatio-temporal transformer for 3d human motion prediction. In *2021 International Conference on 3D Vision (3DV)*, pages 565–574. IEEE, 2021. 3

[5] S. Aliakbarian, P. Cameron, F. Bogo, A. Fitzgibbon, and T. J. Cashman. Flag: Flow-based 3d avatar generation from sparse observations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13253–13262, 2022. 1, 2, 3, 7, 8

[6] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020. 8

[7] J. Chai and J. K. Hodgins. Performance animation from low-dimensional control signals. In *ACM SIGGRAPH 2005 Papers*, pages 686–696. 2005. 2

[8] V. Choutas, F. Bogo, J. Shen, and J. Valentin. Learning to fit morphable models. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VI*, pages 160–179. Springer, 2022. 1, 3

[9] A. Dittadi, S. Dziadzio, D. Cosker, B. Lundell, T. J. Cashman, and J. Shotton. Full-body motion from a single head-mounted device: generating smpl poses from partial observations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11687–11697, 2021. 1, 2, 3, 7, 8

[10] Eyes. Japan co. ltd. eyes. *http://mocapdata.com*, 2018. 8

[11] T. Flash and N. Hogan. The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of neuroscience*, 5(7):1688–1703, 1985. 7

[12] O. A. C. C. for the Arts and Design. Accad. *https://accad.osu.edu/research/motion-lab/systemdata*. 8

[13] C. graphics lab. Cmu graphics lab motion capture database. *http://mocap.cs.cmu.edu/*, 2000. 7, 8

[14] Y. He, R. Yan, K. Fragkiadaki, and S.-I. Yu. Epipolar transformers. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 7779–7788, 2020. 3

[15] Y. Huang, M. Kaufmann, E. Aksan, M. J. Black, O. Hilliges, and G. Pons-Moll. Deep inertial poser: Learning to reconstruct human pose from sparse inertial measurements in real time. *ACM Transactions on Graphics (TOG)*, 37(6):1–15, 2018. 2, 3

[16] J. Jiang, P. Streli, H. Qiu, A. Fender, L. Laich, P. Snape, and C. Holz. Avatarposer: Articulated full-body pose tracking from sparse motion sensing. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part V*, pages 443–460. Springer, 2022. 1, 2, 3, 5, 6, 7, 8, 9

[17] Y. Jiang, Y. Ye, D. Gopinath, J. Won, A. W. Winkler, and C. K. Liu. Transformer inertial poser: Attention-based real-time human motion reconstruction from sparse imus. *arXiv preprint arXiv:2203.15720*, 2022. 2, 3

[18] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations*, 2014. 7

[19] N. Kolotouros, G. Pavlakos, D. Jayaraman, and K. Daniilidis. Probabilistic modeling for human mesh recovery. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11605–11614, 2021. 8

[20] W. Li, H. Liu, H. Tang, P. Wang, and L. Van Gool. Mhformer: Multi-hypothesis transformer for 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13147–13156, 2022. 3

[21] K. Lin, L. Wang, and Z. 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. 3

[22] K. Lin, L. Wang, and Z. Liu. Mesh graphormer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12939–12948, 2021. 3

[23] M. Loper, N. Mahmood, and M. J. Black. Mosh: motion and shape capture from sparse markers. *ACM Trans. Graph.*, 33(6):220–1, 2014. 8

[24] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015. 3

[25] H. Ma, Z. Wang, Y. Chen, D. Kong, L. Chen, X. Liu, X. Yan, H. Tang, and X. Xie. Ppt: token-pruned pose transformer for monocular and multi-view human pose estimation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part V*, pages 424–442. Springer, 2022. 3, 8

[26] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5442–5451, 2019. 2, 7, 8

[27] C. Mandery, Ö. Terlemez, M. Do, N. Vahrenkamp, and T. Asfour. The kit whole-body human motion database. In *2015 International Conference on Advanced Robotics (ICAR)*, pages 329–336. IEEE, 2015. 8

[28] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber. Mocap database hdm05. *Institut für Informatik II, Universität Bonn*, 2(7), 2007. 7, 8

[29] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. Osman, D. Tzionas, and M. J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10975–10985, 2019. 8

[30] J. Qin, Y. Zheng, and K. Zhou. Motion in-betweening via two-stage transformers. *ACM Transactions on Graphics (TOG)*, 41(6):1–16, 2022. 3

[31] D. Rempe, T. Birdal, A. Hertzmann, J. Yang, S. Sridhar, and L. J. Guibas. Humor: 3d human motion model for robust pose estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11488–11499, 2021. 8

[32] RootMotion. Final ik. *https://assetstore.unity.com/packages/tools/animation/final-ik-14290*, 2018. 7, 8

[33] L. Sigal, A. O. Balan, and M. J. Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International journal of computer vision*, 87(1-2):4, 2010. 8

[34] G. Tevet, S. Raab, B. Gordon, Y. Shafir, D. Cohen-Or, and A. H. Bermano. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*, 2022. 3, 6

[35] N. F. Troje. Decomposing biological motion: A framework for analysis and synthesis of human gait patterns. *Journal of vision*, 2(5):2–2, 2002. 7, 8

[36] M. Trumble, A. Gilbert, C. Malleson, A. Hilton, and J. Collomosse. Total capture: 3d human pose estimation fusing video and inertial sensors. In *Proceedings of 28th British Machine Vision Conference*, pages 1–13, 2017. 8

[37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3, 5

[38] T. Von Marcard, B. Rosenhahn, M. J. Black, and G. Pons-Moll. Sparse inertial poser: Automatic 3d human pose estimation from sparse imus. In *Computer graphics forum*, volume 36, pages 349–360. Wiley Online Library, 2017. 2, 3

[39] A. Winkler, J. Won, and Y. Ye. Questsim: Human motion tracking from sparse sensors with simulated avatars. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–8, 2022. 1, 2, 3

[40] D. Yang, D. Kim, and S.-H. Lee. Lobstr: Real-time lower-body pose prediction from sparse upper-body tracking signals. In *Computer Graphics Forum*, volume 40, pages 265–275. Wiley Online Library, 2021. 2, 3, 7, 8

[41] X. Yi, Y. Zhou, M. Habermann, S. Shimada, V. Golyanik, C. Theobalt, and F. Xu. Physical inertial poser (pip): Physics-aware real-time human motion tracking from sparse inertial sensors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13167–13178, 2022. 2

[42] X. Yi, Y. Zhou, and F. Xu. Transpose: Real-time 3d human translation and pose estimation with six inertial sensors. *ACM Transactions on Graphics (TOG)*, 40(4):1–13, 2021. 2, 6, 7

[43] Y. Yuan, J. Song, U. Iqbal, A. Vahdat, and J. Kautz. Physdiff: Physics-guided human motion diffusion model. *arXiv preprint arXiv:2212.02500*, 2022. 7

[44] J. Zhang, Z. Tu, J. Yang, Y. Chen, and J. 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*, pages 13232–13242, 2022. 3, 5

[45] C. Zheng, S. Zhu, M. Mendieta, T. Yang, C. Chen, and Z. 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. 3

[46] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2019. 3