HMD-NeMo: Online 3D Avatar Motion Generation From Sparse Observations

Sadegh Aliakbarian Fatemeh Saleh David Collier Pashmina Cameron Darren Cosker

Microsoft Mixed Reality & AI Lab, Cambridge, UK

Abstract

Generating both plausible and accurate full body avatar motion is the key to the quality of immersive experiences in mixed reality scenarios. Head-Mounted Devices (HMDs) typically only provide a few input signals, such as head and hands 6-DoF. Recently, different approaches achieved impressive performance in generating full body motion given only head and hands signal. However, to the best of our knowledge, all existing approaches rely on full hand visibility. While this is the case when, e.g., using motion controllers, a considerable proportion of mixed reality experiences do not involve motion controllers and instead rely on egocentric hand tracking. This introduces the challenge of partial hand visibility owing to the restricted field of view of the HMD. In this paper, we propose the first unified approach, HMD-NeMo, that addresses plausible and accurate full body motion generation even when the hands may be only partially visible. HMD-NeMo is a lightweight neural network that predicts the full body motion in an online and real-time fashion. At the heart of HMD-NeMo is the spatio-temporal encoder with novel temporally adaptable mask tokens that encourage plausible motion in the absence of hand observations. We perform extensive analysis of the impact of different components in HMD-NeMo and introduce a new state-of-the-art on AMASS dataset through our evaluation.

1. Introduction

Mixed reality technology opens up new means of communication and interaction between people. With *people* at the heart of this technology, generating faithful and believable avatar motion is key to the quality of immersive experiences. Despite great advances in this area, generating full body avatar motion given HMD signal remains a challenge: in many current solutions, avatars only have upper bodies.

Prior works attempted to generate full body avatar motion given sparse or partial observations, such as images [17, 4, 33], 2D joints/keypoints [21, 5], markers [35, 18, 34, 11], and IMUs [28, 14, 32, 31]. While such observations are considered *partial* or *sparse*, they provide much richer input signal compared to a typical HMD's head and hand 6-DoF. More recently, great progress has been made to generate full body motion given only a HMD signal [1, 30, 22, 15], however, they all rely on the availability and full visibility of both hands. While this is the case when, e.g., using motion controllers, many mixed reality experiences do not involve motion controllers and instead rely on hand tracking. This introduces the challenge of partial hand visibility owing to the restricted field of view (FoV) of the HMD sensors.

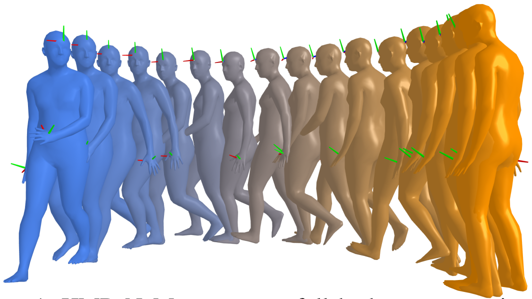
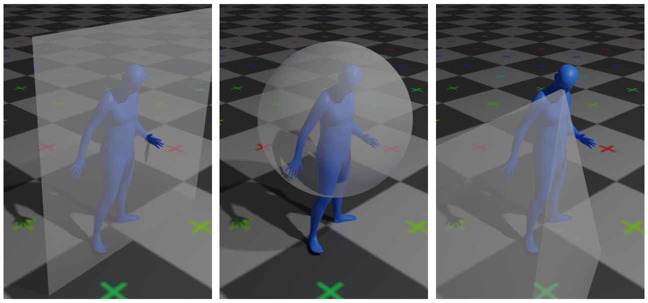


Figure 1. HMD-NeMo generates full body avatar motion given HMD signals, i.e., head and hand 6-DoFs, from hand tracking signal from the HMD as well as hand motion controllers.

In this paper, we address this problem via HMD-NeMo (a **ne**ural **mo**tion model of human given **HMD** signal). Within a unified framework, HMD-NeMo generates full body motion in real time, regardless of whether hands are fully or partially observed, or not observed at all. Our approach is built upon recurrent neural networks to efficiently capture temporal information, and a transformer to capture complex relations between different components of the input signal. At the heart of our approach is the TAMT (**t**emporally **a**daptable **m**ask **t**oken) module, allowing us to handle missing hand observations.

Our contributions are: (1) The first full body avatar motion generation approach capable of generating accurate and plausible motions with full or partial hand visibility. This is a step forward for unlocking fully immersive experiences in mixed reality with fewer limitations on the hardware. (2) Temporally adaptable Mask Tokens (TAMT), a simple yet



(a) Plane FoV (b) Motion Controllers (MC) (c) Hand Tracking (HT)

Scenario	Left hand visibility	Right hand visibility
Motion Controllers	100%	100%
Hand Tracking	53.45%	48.94%

Figure 2. **Top**: Definition of FoV for different scenarios for the same pose. (a) Planar FoV, as used in [8], wherein avatar's left hand is visible while the right hand is not. (b) Fully visible, as in motion controller scenarios, wherein both hands are always visible. (c) HMD's hand tracking camera's FoV, as in hand tracking scenarios, wherein avatar's right hand is visible while the left hand is not. **Bottom**: Hand visibility statistics on AMASS test set.

effective strategy for handling missing hand observations in a temporally coherent way. (3) Extensive experiments and ablations of our method as well as a new state-of-the-art performance on the challenging AMASS dataset.

2. Related Work

Over the past few years, different solutions have been proposed to the problem of full body pose generation given sparse or partial observations, such as markers [35, 18, 34, 11], images [17, 4, 33], 2D joints/keypoints [21, 5], Inertial Measurement Units (IMUs) [28, 14, 32, 31], and HMDs [2, 1, 30, 22, 15]. Among these works, the ones utilizing wearable devices are closer to our approach and thus we discuss them here. Recently, different techniques have been proposed for human pose reconstruction using a sparse set of IMUs attached to the body [28, 14, 32, 31]. While attempts have been made to come up with a minimum number of IMUs for generating full body motion, typically an IMU sensor is used near pelvis, making the problem relatively easy compared to HMD scenarios where the root/pelvis signal is not available. Whilst external sensors, e.g., IMUs [28, 14, 32, 31] and cameras [23] are effective, they are often not as accessible as wearing a HMD. Using only a HMD is desirable from a usability point of view, but generating realistic and faithful human representations from such inputs remains technically challenging.

Given HMD signals, [2] and [8] generate full body poses. While they generate expressive poses faithful to the HMD signal, these approaches [2, 8] only predict static poses, lacking the temporal consistency required for avatar motion generation. While [8] generates poses in the world space, [2] predicts the pose relative to the root. Thus [2] assumes a known root (pelvis joint) as an additional signal. The assumption of known root joint also appears in [9], wherein, unlike [2, 8], the method generates full body *motion* given the HMD signal with Variational Autoencoders [16]. Note that, while [9] generates temporally plausible motions, it works in an offline fashion, predicting the motion for an entire sequence only after observing the whole sequence of HMD signals.

Recently, different techniques have been proposed to generate full body avatar locomotion (in the world coordinates) given HMD signals. In this context, [1] proposes a matching algorithm, aiming to sample closest poses from a motion capture library at sparse time-steps and interpolate between poses. While this guarantees the selection of realistic poses, the output is always limited to the utilized motion capture library. In another work, [30] uses combination of an inverse kinematic (IK) solver and a recurrent neural network to generate upper body and lower body motions, respectively. Combination of different components to solve lower and upper body separately has also been explored in [22], wherein a neural network is trained to predict the root orientation given the HMD signal, which then is used as the feature vector for a motion matching algorithm [6, 13] to generate full body. More recently, fully learning-based approaches have shown promise in generating full body avatar motion [29, 15]. In this context, [29] simulates plausible and physically feasible motions within a reinforcement learning framework and [15] uses a transformer-based approach to generate full body motion given HMD signals.

Although great progress has been made by recent approaches, they all solve the problem of motion generation given head and *both hands*, typically captured with motion controllers. However many mixed reality experiences do not have motion controllers available and instead rely on hand tracking from HMD mounted sensors (e.g., cameras). This introduces the challenge of hand tracking failures and partial hand visibility owing to the restricted field of views. To the best of our knowledge, despite its usability, motion generation in the presence of partial hand observation has not been well-explored. In this paper, we propose a method capable of generating high fidelity and plausible full body motion even in presence of partially visible hands.

3. Proposed Method

In this section, we first define the problem, the scenarios we consider, as well as the input and the desired output representation. We then present our proposed method.

3.1. Problem Definition

Task. The task is to generate full-body 3D human locomotion (predicting both the instantaneous pose and the global trajectory of the human) given the sparse HMD signal in

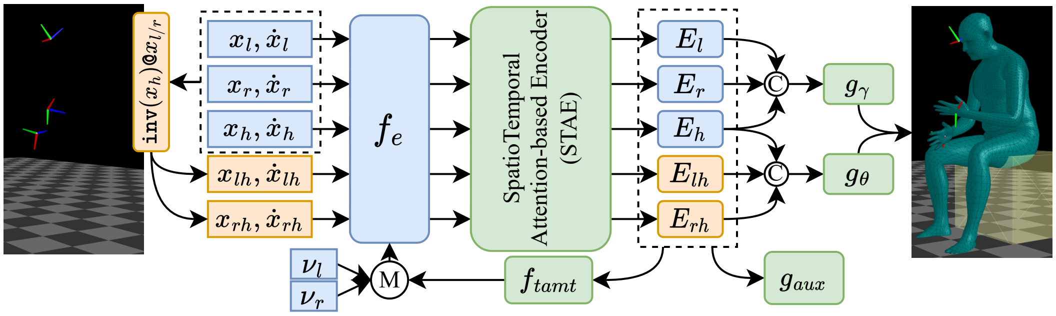


Figure 3. Overview of HMD-NeMo: At each time-step, the model gets as input the HMD signal (left), as in Eq. 1. Such input is the mapped to an embedding space via f_e, acting as the input to the spatio-temporal encoder module STAE. The resulting feature representation is then utilized by two autoregressive decoders, $g\gamma$ and g_θ to predict the global trajectory as well as the pose, respectively. In case the model does not observe a hand (in HT scenario), f_{tamt} fills in the representation of the unobserved hand effectively.

an *online* fashion[1]. That is, given the input signal x_t at each time-step t, the system should predict the 3D human pose and trajectory y_t near-instantaneously. HMD-NeMo achieves this using a neural network parameterized by ϕ.

Scenarios. In this paper, we consider two scenarios: *Motion Controllers (MC)* scenario, wherein hands are *always* tracked via motion controllers using constellation tracking, as illustrated in Fig. 2 (b), and *Hand Tracking (HT)* scenario, wherein hands are tracked via a visual hand tracking system whenever the hands are inside the FoV of the device, as illustrated in Fig. 2 (c). The FoV of the device is defined as a frustum determined by the HMD's hand tracking camera placement and parameters. In this paper, we use a similar FoV to that of the Microsoft HoloLens 2 [26]. Note that, while the MC scenario has been explored recently [1, 30, 22, 15], to the best of our knowledge, this work constitutes the first approach that tackles both MC and HT scenarios within one unified framework[2]. The HT scenario is particularly challenging as hands tend to be out of FoV almost 50% of the time, according to the statistics presented in the table in Fig. 2.

Input Representation. The input signal x^t contains the head 6-DoF $x_h \in \mathbb{R}^{(6+3)}$, the left hand 6-DoF $x_l^t \in \mathbb{R}^{(6+3)}$, and the right hand 6-DoF $x_r^t \in \mathbb{R}^{(6+3)}$, all in the world space. We use the 6D representation to represent the rotations [36]. We additionally provide the hand representations in the head space, $x_{lh}^t \in \mathbb{R}^{(6+3)}$ and $x_{rh}^t \in \mathbb{R}^{(6+3)}$. In the HT scenario, hands may go in and out of FoV of the HMD, so we also provide HMD-NeMo with the hand visibility status for both left and right hand, ν_l^t and ν_r^t, as binary values, 1 being visible and 0 otherwise. Finally, for all 6-DoF signals, we provide the velocity of changes between two consecutive frames. Specifically, for translations we consider $vel(\mathcal{T}^t, \mathcal{T}^{t-1}) = \mathcal{T}^t - \mathcal{T}^{t-1}$, where $\mathcal{T}$ is the translation, and for rotations we consider the geodesic changes in the rotation $vel(R^t, R^{t-1}) = (R^{t-1})^{-1}R^t$, where R is the rotation, which together constitute the velocity 6-DoF $\dot{x}^t$. Overall, the input to the HMD-NeMo, $x^t \in \mathbb{R}^{92}$, can be written as

$$x^t = \{x_h, x_l, x_r, x_{lh}, x_{rh}, \dot{x}_h, \dot{x}_l, \dot{x}_r, \dot{x}_{lh}, \dot{x}_{rh}, \nu_l, \nu_r\}^t \tag{1}$$

Output Representation. The output of HMD-NeMo contains the pose (including the root orientation), $\theta^t \in \mathbb{R}^{J\times 3}$ represented with axis-angle rotations for the J joints in the body, and the global position in the world, $\gamma^t \in \mathbb{R}^3$ represented as the root position, resulting in $y^t \in \mathbb{R}^{(J+1)\times 3}$,

$$y^t = \{\theta, \gamma\}^t \tag{2}$$

The sequence of $\theta^{0:T}$ and $\gamma^{0:T}$ then represent the avatar motion as well as its global trajectory for the period $[0, T]$. Note that, in this paper, we may drop the time superscript t for better readability and include it when necessary.

3.2. HMD-NeMo

Overview. HMD-NeMo's pipeline is illustrated in Fig. 3. As described in Section 3.1, the model gets as input the information about the head and hands in world coordinate system, the hands expressed in the head space, as well as their velocities, as described in Eq. 1. To express hands in head space, we consider $x_{lh} = x_h^{-1}x_l$ (similarly, $x_{rh} = x_h^{-1}x_r$). This representation then acts as the input to an embedding layer, f_e, which aims to (1) map the raw input to an embedding space and (2) handle the unobserved hands. Given the input f_e, the next module, STAE, learns (a) how each representation evolves over time and (b) how different components of the input, i.e, head and hands, are correlated. Once such a rich representation is obtained, two auto-regressive decoders, g_θ and g_γ, generate the body pose and the global

[1] Note that, similar to existing work on HMD-driven motion generation [1, 15, 9], our approach does not predict body shape parameters and only focuses on the poses.

[2] Note that [8] also considers a form of partial hand visibility by defining a plane FoV, wherein, as illustrated in Fig. 2 (a), joints in front of head is are considered visible. However, this scenario does not apply to practical use cases.

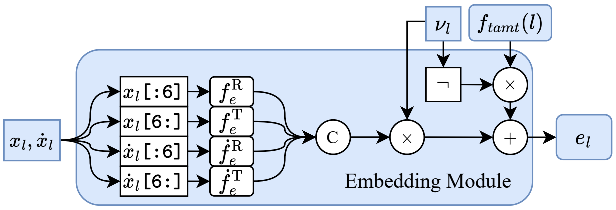


Figure 4. Overview of the embedding module. Note that x_l is only provided as an example and this module applies to all 6-DoF inputs appearing in x^t as in Eq. 1.

position of the avatar, respectively. At each time step, the output of STAE is used to update the mask tokens (described below) as a representation for the hand signals that may be missing in the next time-step. To aid training, we also include an auxiliary task of human pose reconstruction in SE(3), denoted by g_{aux}.

Head and hand embedding (f_e). As shown in Fig. 4, the embedding module f_e gets as input the head and hands 6-DoFs and velocities and maps them to a higher-dimensional embedding space. As the range of values corresponding to the rotations is different from those of the translations, we decouple such information and embed them via separate shallow MLPs and concatenate the results back together. For instance, for the observed left hand in the world coordinate system, the embedding representation is

$$e_l^{visible} = \left[f_e^{\mathrm{R}}(x_l[:6]), f_e^{\mathrm{T}}(x_l[6:]), \dot{f}_e^{\mathrm{R}}(\dot{x}_l[:6]), \dot{f}_e^{\mathrm{T}}(\dot{x}_l[6:]) \right] \tag{3}$$

where f_e^{R} and f_e^{T} are MLPs responsible for computing the rotation and translation embedding, acting on the first 6 elements (the 6D rotation representation) and the last 3 elements (the translation) of the input, respectively (similarly for $\dot{f}_e^{\mathrm{R}}$ and $\dot{f}_e^{\mathrm{T}}$ which act on velocities).

In the HT scenario, hands may not be visible to the model, hence computing such embedding representation is not possible. Thus, given the status of ν_l and ν_r, the embedding module decides to either compute the embedding or utilize the output of the f_{tamt} (described below), a set of temporally adaptable mask tokens, instead of a missing hand observation (denoted by M in Fig. 3). As illustrated in Fig 4, the embedding of the left hand in the world coordinate system can be computed as

$$e_l = \nu_l e_l^{visible} + (1 - \nu_l) f_{tamt}(l). \tag{4}$$

Spatio-temporal encoder (STAE). The output of f_e on each component of the input stream is a non-temporal feature, computed independent of other components in the input. While an expressive representation of each component, it lacks temporal and spatial correlation information. We specifically care about these characteristics because the model is required to generate temporally coherent motion and also because the motion of one body part often impacts or determines the motion of other body part(s). To obtain a more informative representation from the head and hands, HMD-NeMo first learns the isolated temporal features of each component of input representation and then learns how they are spatially correlated [27].

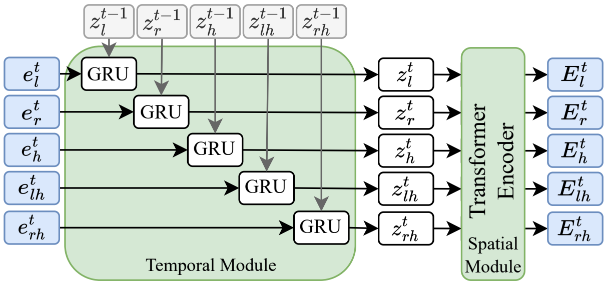


Figure 5. Overview of the spatio-temporal encoder module.

As illustrated in Fig. 5, to learn the temporal representation of the input signal, we use gated recurrent units (GRUs) [7]. With a GRU module on top of each component in the input, the model learns how each component, e.g., head, evolves over time, independent of other components in the input. This information is compressed in the hidden state of the GRU cell, z, which is then utilized to learn how different components in the input relate to each other. This is achieved by using a standard transformer encoder on the GRU hidden states, thanks to the self-attention mechanism.

Temporally adaptable mask tokens (TAMTs). As discussed in Section 3.1, hands may not be visible to the model, and thus, there is no representative input signal for the f_e module. To address this issue, in case of a missing hand observation, our model produces a feature vector, f_{tamt}, to represent the missing hand observation. To compute f_{tamt}, as illustrated in Fig. 6, we use the output of STAE for the hand observation that may be missing in the next time step as well as the output of STAE for the head. Note that head joint is the reference joint and is always available. The combination of these two features is a rich representation of the past state of the missing hand signal (both temporally and spatially); this is then used to compute the f_{tamt}. In order to encourage f_{tamt} to learn information about the missing hand observation, as illustrated by the *Forecaster* module in Fig. 6, we introduce a forecasting auxiliary task to forecast the state (6-DoF) of the corresponding hand in the next time-step.

3.3. Training HMD-NeMo

To train HMD-NeMo, we rely on the availability of a motion capture dataset, represented as SMPL [19] parameters (pose, shape, and global trajectory). For each sequence, we then simulate the HMD on the subject. In case of HT, we also simulate a FoV frustum to be able to model the hand visibility status (ν_l and ν_r). We then train HMD-NeMo with a loss function of the form

$$\mathcal{L} = \alpha_{data}\mathcal{L}_{data} + \alpha_{smooth}\mathcal{L}_{smooth} \qquad (5)$$
$$+\alpha_{SE(3)}\mathcal{L}_{SE(3)} + \alpha_{forecast}\mathcal{L}_{forecast} + \alpha_{aux}\mathcal{L}_{aux}$$

The data loss term is the squared error between the predicted pose and trajectory and those of the ground truth,

$$\mathcal{L}_{data} = \sum_{t=1}^{T} ||\hat{\theta}^t - \theta^t||_2^2 + ||\hat{\gamma}^t - \gamma^t||_2^2 \qquad (6)$$

Note that, in practice, the pose decoder has two heads, one each for predicting the body pose and the global root orientation. To further enhance the temporal smoothness, we penalize the discrepancy between the velocity of changes in the prediction to that of the ground truth

$$\mathcal{L}_{smooth} = \sum_{t=2}^{T} ||\delta\hat{\theta}^t - \delta\theta^t||_1 + ||\delta\hat{\gamma}^t - \delta\gamma^t||_1 \qquad (7)$$

where $\delta\hat{\theta}^t = \hat{\theta}^t - \hat{\theta}^{t-1}$ ($\delta\hat{\gamma}$, $\delta\theta$, and $\delta\gamma$ follow similarly). In addition to computing the reconstruction loss on the SMPL parameters, i.e., to relative joint rotations, we found it extremely useful to also utilize the reconstruction loss of each joint transformation independent of its parent, i.e., in the world space. To compute this reconstruction loss, we use the SMPL model to compute the joint transformations in SE(3) given the predicted and ground truth pose and trajectory parameters. Thus, the SE(3) reconstruction loss is

$$\mathcal{L}_{SE(3)} = \sum_{t=1}^{T} ||\hat{P}_{SE(3)}^t - P_{SE(3)}^t||_2^2 \qquad (8)$$

where $P_{SE(3)}$ is the body pose in SE(3). The next loss term corresponds to the forecasting auxiliary task in the TAMT module, where the goal is to minimize the distance between the predicted next hand and the ground truth next hand,

$$\mathcal{L}_{forecast} = \sum_{t=2}^{T} \sum_{j\in\{l,r\}} ||\hat{x}_j^t - x_j^t||_2^2 \qquad (9)$$

Finally, we have our loss term for the auxiliary task, aiming to minimize the predicted full body joint transformations from STAE's features, $\hat{P}_{aux}$, to the ground truth body joint transformations,

$$\mathcal{L}_{aux} = \sum_{t=1}^{T} ||\hat{P}_{aux}^t - P_{SE(3)}^t||_2^2 \qquad (10)$$

Our model is trained with αs all being set to 1 in Eq. 6.

3.4. Optimization

Once trained, HMD-NeMo is capable of generating high fidelity and plausible human motion given only the HMD signal. However, as is typical of learning-based approaches, the direct prediction of the neural network does not precisely match the observations i.e., the head and hands, even if it is perceptually quite close. To close this gap between the prediction and the observation, optimization can be used. This adjusts the pose parameters to minimize an energy function of the form $\mathcal{E} = \mathcal{E}_{data} + \mathcal{E}_{reg}$, where $\mathcal{E}_{data}$ is the energy term that minimizes the distance between the predicted head and hands to the observed ones, and $\mathcal{E}_{reg}$ is additional regularization term(s). To define the data energy term, we define the residual $\mathcal{R} = \sum_{j\in\{h,l,r\}}(x_j - \hat{x_j})$, i.e., the difference between the predicted head/hand joint to that of the observation. Given $\mathcal{R}$, a typical, non-robust data energy term could be written as $\mathcal{E}_{nr} = \mathcal{R}^2$, i.e., the L2 loss. This suits the MC scenario perfectly, where head, left hand, and right hand are *always* available. But this energy term may be misleading in HT scenario where hands are going into and out of FoV often, and thus leading to abrupt pose changes when hands appear back in the FoV (see supp mat for further details). To remedy this issue, we utilize a more robust [3] alternative to the data energy term,

$$\mathcal{E}_r(\mathcal{R}, a, b, c) = b\frac{|a-2|}{a}\left(\left(\frac{(\frac{\mathcal{R}}{c})^2}{|a-2|} + 1\right)^{(\frac{a}{2})} - 1\right) \quad (11)$$

where a, b, and c are hyper-parameters that determine the shape of the loss (see supp mat for further details). Unlike $\mathcal{E}_{nr}$, $\mathcal{E}_r$, considers large discrepancies between the prediction and observation as outliers, not penalizing the prediction strongly and thus does not push the prediction to move toward the observation. Thus abrupt changes in the arm poses are avoided and optimization stays on course despite large variation in the velocity metric (when caused by hand visibility changes). Of course, utilizing Eq. 11 adversely affects the fidelity, but a trade-off between the plausibility and fidelity can be chosen to suit the application of interest.

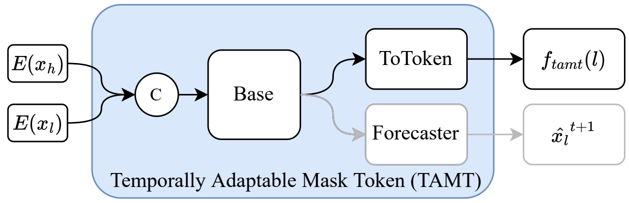


Figure 6. Overview of TAMT module. Note that here we illustrate TAMT of x_l as an example, the same applies f to both hands in both head space and world space.

Note that, since all observations relate to the upper body, during optimization we only optimize the upper body pose parameters and global root trajectory, while keeping the predicted lower body untouched.

4. Experiments

In this section, we introduce the dataset and metrics, with implementation details in the supplementary material. We then present the experimental results and ablation studies.

Dataset. We follow the recent common practice [15, 24, 8, 9] of using a subset of AMASS [20] for training and evaluation. AMASS is a large collection of human motion se-

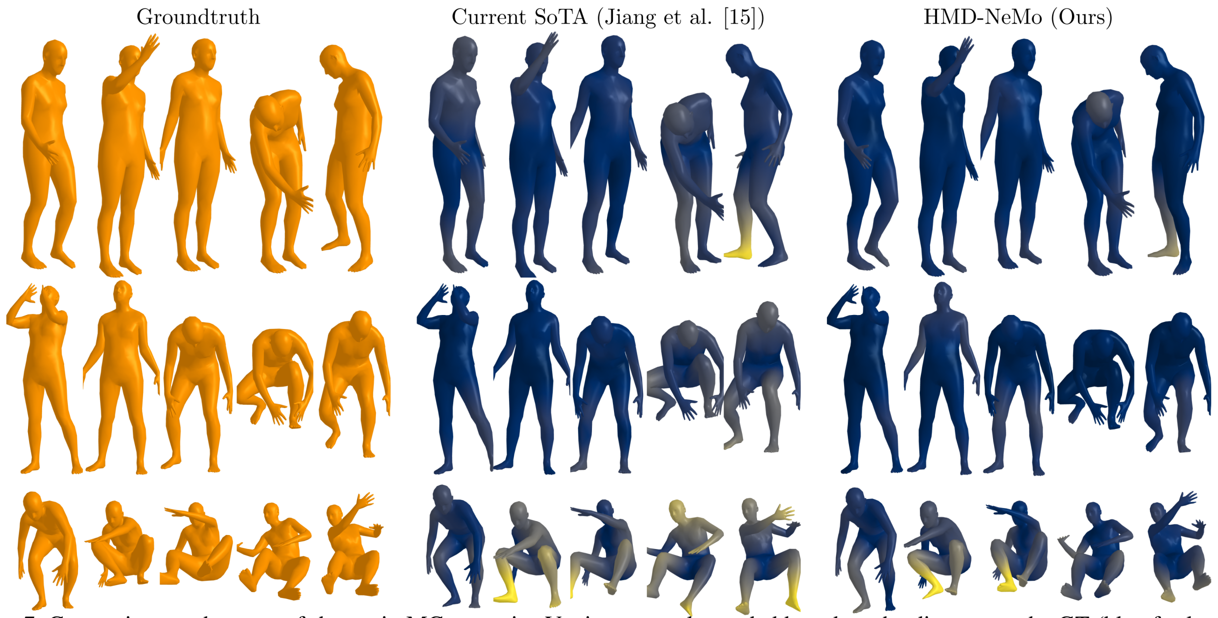


Figure 7. Comparison to the state-of-the-art in MC scenario. Vertices are color-coded based on the distance to the GT (blue for low error and yellow for high error). Last row depicts a hard example with complex body pose and motion.

quences, converted to 3D human meshes in the SMPL [19] representation, wherein every motion sequence contains information about poses θ and the global trajectory γ. To synthesize the HMD scenario, we compute the global transformation matrices for the head and hands as input. In the case of hand tracking, we define a FoV for the HMD and mask out the hands whenever they are out of FoV. To make a fair comparison, for both training and evaluation, we follow the splits suggested by [15].

Metrics. To evaluate the performance of our approach as well as the competing baselines, we report the mean per-joint position error (MPJPE [cm]) and the mean per-joint velocity error (MPJVE [cm/s]). We compare our approach with recent techniques [15, 1, 30, 9][3]. As these approaches do not tackle the hand tracking scenario, we compare HMD-NeMo against them for the motion controller scenario. We evaluate HMD-NeMo for hand tracking scenarios separately. In our experiments, we do not use the ground truth body shape parameters, but instead use the same default shape for all sequences. This follows the evaluation used in previous work [1, 15, 9].

4.1. Comparison to the state-of-the-art

We compare HMD-NeMo with existing approaches that tackle the problem of full-body motion generation given HMD signal. To the best of our knowledge, all existing approaches only tackle the MC scenario, and thus we compare them with this setting[4]. As demonstrated in Table 1, HMD-NeMo is not only more accurate (lower MPJPE), but also generates smoother motion with joint velocities similar to that in the ground truth (lower MPJVE), introducing a new state-of-the-art on the AMASS dataset for both metrics. Note that [25] has the highest errors, as shown in Table 1 as it only optimizes the pose of the head and hands and ignores the accuracy and smoothness of the rest of the joints. Consequently, since [30] utilizes FinalIK within its framework, its performance becomes bounded by the quality of FinalIK. The learning-based approaches [15, 9], however, perform much better than [25, 1, 30], highlighting the value of data-driven methods trained on large-scale motion capture datasets. In Fig. 7, we compared HMD-NeMo with the second best performing baseline [15].

Method	MPJPE ↓	MPJVE ↓
FinalIK [25]	18.09	59.24
Ahuja et al. [1]	7.83	100.54
Yang et al. [30]	9.02	44.97
Dittadi et al. [9]	6.83	37.99
Jiang et al. [15]	4.18	29.40
HMD-NeMo (Ours)	**1.90**	**24.99**

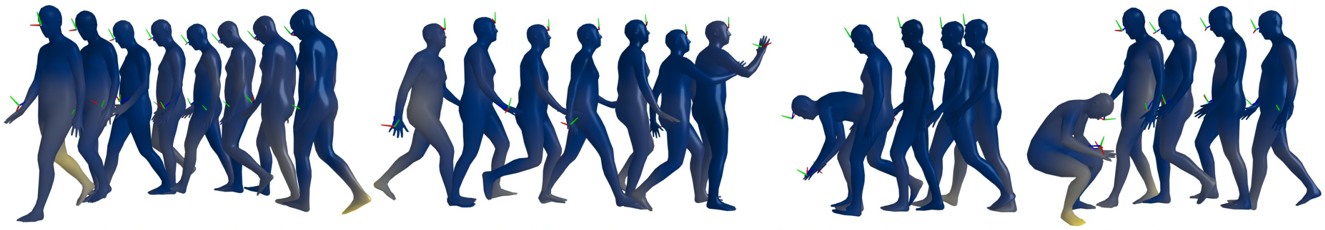
Table 1. Comparison to the state-of-the-art approaches in MC scenario, where both hands are always visible.

4.2. Ablation Studies

In this section, we comprehensively evaluate different aspects of HMD-NeMo, including evaluation in HT scenario, cross-dataset evaluations, ablation studies on input signals, model architectures, and loss terms. We also ana-

[3] The approach [9] is modified to predict motion in world coordinates.

[4] Note, we exclude very recent approaches [22, 29] since they either do not report on AMASS or the code is not publicly available.



(a) Walk in a semi-circle

(b) Walk, then knock

(c) Walking, then put sth down

(d) Walk, then sit

Figure 8. Qualitative results in HT scenario. Vertices are color-coded based on the distance to the GT (blue for low error and yellow for high error). See the supplementary video for more qualitative results.

Model	Mask Type	MPJPE ↓	MPJVE ↓
HMD-NeMo	Learned & fixed	5.59	42.81
HMD-NeMo	TAMTs (Ours)	**2.48**	**31.30**

Table 2. Effect of Temporally Adaptable Mask Tokens (TAMTs). These results represent the HMD-NeMo prediction before optimization solely to evaluate the effect of the TAMT module.

Method	Test on CMU		Test on BMLrub		Test on HDM05	
	MPJPE ↓	MPJVE ↓	MPJPE ↓	MPJVE ↓	MPJPE ↓	MPJVE ↓
FinalIK [25]	18.82	56.83	17.58	60.64	18.43	62.39
Ahuja et al. [1]	18.77	139.17	13.30	134.77	17.90	140.61
Yang et al. [30]	12.96	49.94	11.00	60.74	11.94	48.26
Dittadi et al. [9]	13.04	51.69	9.69	51.80	10.21	40.07
Jiang et al. [15]	8.37	35.76	7.04	43.70	8.05	30.85
HMD-NeMo (Ours)	**7.13**	**31.23**	**6.46**	**40.38**	**6.80**	**27.91**

Table 3. Results of cross-dataset evaluation between different methods. In order to compare with existing methods, we provide results in MC scenario.

lyze the effect optimization in both HT and MC scenarios. Additionally, we provide results of HMD-NeMo in both HT and MC scenarios for various body parts, as well as qualitative effect of optimization in the supplementary material.

Evaluation for the hand tracking scenario. To the best of our knowledge, HMD-NeMo is the first approach to address the problem of human motion generation given *partial* HMD signal, applicable to HT scenario. We believe that TAMT module is the key to the success of HMD-NeMo in handling missing/partial observation, so we compare it with an alternative commonly used in the Vision Transformers [10, 12], which uses a learned set of parameters (i.e., `nn.Parameters` in PyTorch) to model the missing observations[5]. While learned parameters are fixed after training for every data point and every sequence, TAMT temporally updates itself at each time-step given the current state of the model. Table 2 (especially MPJVE) shows the superiority of TAMT module over learned and fixed parameters in handling missing hand observations. Qualitatively, Fig. 8 illustrates how HMD-NeMo performs in HT scenario.

Cross-dataset evaluation. To investigate the generalizability of HMD-NeMo, we conduct a 3-fold cross-dataset evaluation as in [15], wherein the models are trained on two subsets and test on the other subset. In order to compare our approach with existing methods, we conduct this experiment in MC scenario. As shown in Table 3, HMD-NeMo outperforms existing approaches in all three datasets, by a considerable margin, highlighting its generalizability.

Evaluating the effect of input signal. As discussed in Section 3.1, HMD-NeMo utilizes head and hands, hands in the head space, as well as the corresponding velocities, as in Eq. 1. Table 4 summarizes the effect of each component in the input signal. As shown, adding hands in the head space on top of head and hands in the world coordinates leads to better pose prediction, and thus reduces the MPJPE. Incorporating the velocities has a significant contribution to generating more temporally coherent motion, and thus reduces the MPJVE considerably. Considering all input signals, as in Eq. 1 leads to best MPJPE and MPJVE.

Input Signal	MPJPE ↓	MPJVE ↓
$x^t = \{x_h, x_l, x_r\}^t$	4.38	39.63
$x^t = \{x_h, x_l, x_r, x_{lh}, x_{rh}\}^t$	3.21	38.32
$x^t = \{x_h, x_l, x_r, \dot{x}_h, \dot{x}_l, \dot{x}_r\}^t$	3.53	35.27
Full input signals (Eq. 1)	**2.48**	**31.30**

Table 4. Evaluating the effect of various input signals in HT scenario. Note that the hand visibility status ν_l, ν_r are always provided to the model.

Evaluating the effect of STAE. One core component of HMD-NeMo is the spatioTemporal attention-based encoder (STAE). Here, we study the design choices of STAE in Table 5. As shown, removing the GRU component, which is responsible to learn the temporal information of various input signals, affects the MPJVE considerably. Removing the transformer encoder, which aims at learning the relation between head and hands, adversely affects the MPJPE. Unsurprisingly, removing the entire STAE module, i.e., going from input embedding layer to the decoders, is a considerably weak baseline, with poor pose quality and temporal coherency. HMD-NeMo utilizes the power and efficiency of GRU module to learn the temporal information as well as the expressively of the transformer encoder to learn how various inputs are related to each other. This design leads to

[5] This can be considered as removing the unobserved hand (similar to [2]) and replacing it with learned and fixed parameters.

STAE Components	MPJPE ↓	MPJVE ↓
No GRU	4.34	46.61
No Transformer	5.55	39.10
No GRU, No Transformer	7.09	52.26
With STAE (Ours)	**2.48**	**31.30**

Table 5. Evaluating the effect of STAE module in HT scenario.

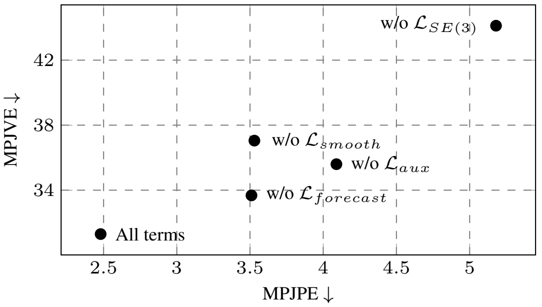


Figure 9. Evaluating the effect of each loss term. Please note that each term is evaluated in isolation.

best performing MPJPE and MPJVE, especially in HT scenario wherein the model faces regular missing observations.

Evaluating the effect of each loss term. As described in Section 3.3 and shown in Eq. 6, HMD-NeMo is trained with five different loss terms. While $\mathcal{L}_{data}$ is the essential term for training HMD-NeMo, other loss terms contribute significantly to the performance of the model. In Fig. 9, we illustrate the contribution of each loss term, in leave-one-term-out manner, on the changes in the MPJPE and MPJVE metrics. $\mathcal{L}_{smooth}$ has a significant impact on improving the MPJVE, while $\mathcal{L}_{aux}$ improves the MPJPE, and $\mathcal{L}_{forecast}$, which acts only on hands mildly improves MPJPE and MPJVE (its contribution to the total error metric is relatively small as it acts on a significantly fewer joints). $\mathcal{L}_{SE(3)}$ makes the largest contribution to the reduction of error in both metrics. It is likely that $\mathcal{L}_{SE(3)}$ aims to bridge that gap[6] between the representation of the input signal (head and hand global transformation matrices) and the representation of the output pose (global root orientation and relative joint rotations).

Evaluating the effect of optimization. While the prediction of HMD-NeMo is very good, if the budget allows, it can be further optimized to improve accuracy (see supp mat for qualitative results). To bridge the gap between the predictions and observations, we optimize the pose prediction from HMD-NeMo, so that it matches the head and hand observations. As described in Section 3.4, in MC scenario, where both hands are always available, a simple optimization (with non-robust data energy term) loop can be used. The effect of such optimization in MC scenario is provided in Table 6. However, considering the same strategy for the HT scenario, where we may partially observe hands, this may not be optimal as we lose plausibility in the generated motions (captured by the MPJVE metric). As described in Section 3.4, depending on the scenario and experience requirements, one may choose to (1) avoid optimizing the predictions if plausibility is the highest priority (first row of Table 7), (2) use non-robust energy term if fidelity is the highest priority (second row of Table 7), or (3) use a robust energy term as a trade-off between fidelity and plausibility (third row of Table 7).

Configuration	MPJPE ↓	MPJVE ↓
Without optimization	2.07	26.07
With optimization	**1.90**	**24.99**

Table 6. Effect of optimization in MC scenario, where both hands are always visible.

Configuration	Priority	MPJPE ↓	MPJVE ↓
No Optimization	Plausibility	2.48	31.30
Opt. with $\mathcal{E}_{nr}$	Fidelity	2.32	33.51
Opt. with $\mathcal{E}_{r}$	Both	2.37	31.33

Table 7. Configuring HT scenario. Our model offers a range of choices to provide the best possible end-user experience despite incomplete observations.

Performance analysis. With 5.3M parameters, at inference time, HMD-NeMo requires only 4.4 ms to generate a pose given a HMD signal on a typical laptop CPU. On a NVIDIA Tesla P100 GPU, our model runs at 265 fps, with 1 iteration of optimization costs 3ms per frame (which could be further improved with more optimized implementation). Such performance makes HMD-NeMo a potential solution for HMD-driven avatar animation in immersive environments.

5. Conclusion

In this paper, we present HMD-NeMo, a unified approach to generate full-body avatar motion in both motion controller and hand tracking scenarios. To handle the unobserved hands in a temporally coherent and plausible manner, we introduce TAMT module. It is worth noting that, in this paper, we considered one major reason for partial hand visibility, i.e., hands appearing out of the FoV, however, in practice, hands may not be visible to the hand tracking camera due to failure in tracking and occlusion by another object or by another body part. While such cases are not considered in the data augmentation in our paper, they certainly can be taken into account and TAMT can be used to fill such gaps with no additional modification; this is left for future exploration. We provide extensive analyses on different components of HMD-NeMo, and shed light on various choices for optimization on top of neural network's predictions when it comes to production priorities (plausibility versus fidelity).

[6] This experiment only evaluates the contribution of each loss term independently. Evaluating the combination of loss terms remains for future investigations.

References

[1] Karan Ahuja, Eyal Ofek, Mar Gonzalez-Franco, Christian Holz, and Andrew 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] Sadegh Aliakbarian, Pashmina Cameron, Federica Bogo, Andrew Fitzgibbon, and Thomas 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.

[3] Jonathan T Barron. A general and adaptive robust loss function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4331–4339, 2019.

[4] Benjamin Biggs, Sébastien Ehrhadt, Hanbyul Joo, Benjamin Graham, Andrea Vedaldi, and David Novotny. 3D multi-bodies: Fitting sets of plausible 3D human models to ambiguous image data. 2020.

[5] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *European conference on computer vision*, pages 561–578. Springer, 2016.

[6] Michael Büttner and Simon Clavet. Motion matching-the road to next gen animation. *Proc. of Nucl. ai*, 2015(1):2, 2015.

[7] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[8] Vasileios Choutas, Federica Bogo, Jingjing Shen, and Julien Valentin. Learning to fit morphable models. *arXiv preprint arXiv:2111.14824*, 2021.

[9] Andrea Dittadi, Sebastian Dziadzio, Darren Cosker, Ben Lundell, Thomas J Cashman, and Jamie 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.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[11] Nima Ghorbani and Michael J Black. Soma: Solving optical marker-based mocap automatically. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11117–11126, 2021.

[12] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.

[13] Daniel Holden, Oussama Kanoun, Maksym Perepichka, and Tiberiu Popa. Learned motion matching. *ACM Transactions on Graphics (TOG)*, 39(4):53–1, 2020.

[14] Yinghao Huang, Manuel Kaufmann, Emre Aksan, Michael J Black, Otmar Hilliges, and Gerard 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.

[15] Jiaxi Jiang, Paul Streli, Huajian Qiu, Andreas Fender, Larissa Laich, Patrick Snape, and Christian Holz. AvatarPoser: Articulated full-body pose tracking from sparse motion sensing. In *Proceedings of European Conference on Computer Vision*. Springer, 2022.

[16] Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *International Conference on Learning Representations, ICLR*, 2014.

[17] Nikos Kolotouros, Georgios Pavlakos, Dinesh Jayaraman, and Kostas Daniilidis. Probabilistic modeling for human mesh recovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11605–11614, 2021.

[18] Matthew Loper, Naureen Mahmood, and Michael J Black. MoSh: Motion and shape capture from sparse markers. *ACM Transactions on Graphics (TOG)*, 33(6):1–13, 2014.

[19] 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.

[20] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, pages 5442–5451, Oct. 2019.

[21] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael 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.

[22] Jose Luis Ponton, Haoran Yun, Carlos Andujar, and Nuria Pelechano. Combining Motion Matching and Orientation Prediction to Animate Avatars for Consumer-Grade VR Devices. *Computer Graphics Forum*, 2022.

[23] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. PIFuHD: Multi-level pixel-aligned implicit function for high-resolution 3D human digitization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 84–93, 2020.

[24] Fatemeh Saleh, Sadegh Aliakbarian, Hamid Rezatofighi, Mathieu Salzmann, and Stephen Gould. Probabilistic tracklet scoring and inpainting for multiple object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14329–14339, 2021.

[25] Unity Asset Store. Final IK. `https://assetstore.unity.com/packages/tools/animation/final-ik-14290`, 2021.

[26] Dorin Ungureanu, Federica Bogo, Silvano Galliani, Pooja Sama, Casey Meekhof, Jan Stühmer, Thomas J Cashman,

Bugra Tekin, Johannes L Schönberger, Pawel Olszta, et al. Hololens 2 research mode as a tool for computer vision research. *arXiv preprint arXiv:2008.11239*, 2020.

[27] 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.

[28] Timo Von Marcard, Bodo Rosenhahn, Michael J Black, and Gerard 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.

[29] Alexander Winkler, Jungdam Won, and Yuting Ye. Questsim: Human motion tracking from sparse sensors with simulated avatars. *arXiv preprint arXiv:2209.09391*, 2022.

[30] Dongseok Yang, Doyeon Kim, and Sung-Hee 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.

[31] Xinyu Yi, Yuxiao Zhou, Marc Habermann, Soshi Shimada, Vladislav Golyanik, Christian Theobalt, and Feng 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.

[32] Xinyu Yi, Yuxiao Zhou, and Feng Xu. TransPose: Real-time 3D human translation and pose estimation with six inertial sensors. *ACM Transactions on Graphics (TOG)*, 40(4):1–13, 2021.

[33] Andrei Zanfir, Eduard Gabriel Bazavan, Hongyi Xu, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu. Weakly supervised 3D human pose and shape reconstruction with normalizing flows. In *European Conference on Computer Vision*, pages 465–481. Springer, 2020.

[34] Mihai Zanfir, Andrei Zanfir, Eduard Gabriel Bazavan, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu. THUNDR: Transformer-based 3D human reconstruction with markers. *arXiv preprint arXiv:2106.09336*, 2021.

[35] Yan Zhang, Michael J Black, and Siyu Tang. We are more than our joints: Predicting how 3D bodies move. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3372–3382, 2021.

[36] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao 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.