Diffusion Model is a Good Pose Estimator from 3D RF-Vision (Supplementary)

Junqiao Fan[1], Jianfei Yang[1,2⋆], Yuecong Xu[1], and Lihua Xie[1]

[1] School of Electrical and Electronic Engineering
[2] School of Mechanical and Aerospace Engineering
Nanyang Technological University, Singapore
{fanj0019, jianfei.yang, xuyu0014, elhxie}@ntu.edu.sg

Appendix

1 mmWave Human Sensing

In Figure 1, we illustrate the utilization of mmWave radar for human sensing, which detects human actions within a range of 3-5 meters from the radar. This process generates mmWave point clouds (PCs). Simultaneously, a keypoint annotation system such as VICON, Mocap, or Cameras is deployed to record ground-truth human poses for reference. The mmWave radar-based human pose estimation refers to training a neural network to estimate human poses using mmWave radar point clouds as input.

For general mmWave human sensing, FMCW (Frequency Modulated Continuous Wave) chirp signals are transmitted and their reflections are received through antenna arrays. These chirp signals are defined by parameters such as start frequency f_c, bandwidth B, and duration T_c. To generate radar PCs [15], range-FFT separates different frequency components f from the IF signals, enabling the extraction of object distances using the formula $R = \frac{cfT_c}{2B}$, where c is the speed of light. Doppler-FFT measures phase changes ω of the IF signals, facilitating the calculation of object velocities using $v = \frac{\lambda\omega}{4\pi T_c}$, where λ is the wavelength of the chirp. Elevation angles φ and azimuth angles θ of the detected objects are determined based on $\varphi = sin^{-1}(\frac{\omega_z}{\pi})$ and $\theta = sin^{-1}(\frac{\omega_x}{cos(\varphi)\pi})$, where ω_x is the phase change between azimuth antennas and corresponding elevation antennas, and ω_z is the phase change of consecutive azimuth antennas. Finally, the Cartesian coordinates (x, y, z) of the detected point clouds are calculated as follows: $x = Rcos(\varphi)sin(\theta)$, $z = Rsin(\varphi)$, and $y = (R^2 - x^2 - z^2)$.

2 The Diffusion Model for Human Pose Generation

Forward process. The forward process involves the gradual sampling of increasingly noisy human poses, resulting in an intermediate distribution of noisy poses that serve as training guidance. Here, $k \in [1..K]$ denotes the diffusion steps,

⋆ J. Yang is the project lead.

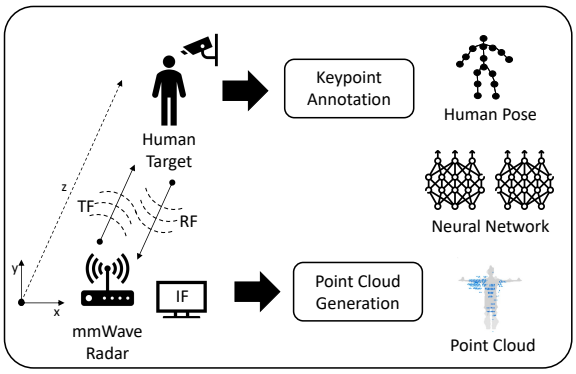


Fig. 1: Workflow of mmWave sensing and mmWave human pose estimation.

where noisy human poses H_k are sampled by adding noise to the original ground truth pose H_0. Following the Markov process, we interatively add noise to the pose H_{k-1} and obtain a noisier version of pose H_k:

$$q\left(H_k \mid H_{k-1}\right) = \mathcal{N}\left(H_k \mid \sqrt{\alpha_k} H_{k-1}, \left(1-\alpha_k\right) I\right), \tag{1}$$

$$H_k = \sqrt{1-\beta_k} H_{k-1} + \beta_k \varepsilon, \tag{2}$$

where β_k denotes the noise scale, $\varepsilon \in \mathcal{N}(0, I)$ denotes the random noise, and $\alpha_k = 1 - \beta_k$. With a small noise scale $\beta_k \in [0.0001, 0.001]$, H_k is approaching to H_{k-1}, which allows us to model both the forward sampling $q(H_k|H_{k-1})$ and the reverse estimation $p_\theta(\hat{H}_{k-1}|H_k, C)$ as Gaussian distributions. To directly sample H_k from the ground truth H_0, Eq. 2 can be rewritten to:

$$H_k = \sqrt{\gamma_k} H_0 + \sqrt{1-\gamma_k}\boldsymbol{\varepsilon}, \tag{3}$$

where $\gamma_k = \prod_{i \in [1..k]} \alpha_k$. Eventually, when $k \to \infty$, H_k approaches pure random noise following Gaussian distribution.

Reverse process. At each iteration k of the reverse process, a cleaner pose $\hat{H}_{k-1}$ is estimated to approximate H_{k-1} (generated by the forward process), given a noisy pose H_k and the conditional set C. The reverse process can be formulated as:

$$p_\theta\left(\hat{H}_{0:K} \mid H_0, C\right) = p\left(H_K\right) \prod_{k=1}^{K} p_\theta\left(\hat{H}_{k-1} \mid H_k, C\right). \tag{4}$$

During training, $H_{1:K}$ are obtained by adding noise to the ground truth H_0, following the forward process. However, during inference, as the ground truth is unavailable, we set $H_{1:K} = \hat{H}_{1:K}$ by iteratively inputting the estimated human poses $\hat{H}_{1:K}$ into the trained diffusion model. As discussed in the main paper, $\hat{H}_K$ is initialized by a coarsely estimated pose $\tilde{H}$ as the starting point of the reverse process.

Table 1: Overview of the datasets used for mmWave human pose estimation.

Dataset:	mmMesh	mmBody	mm-Fi
Radar Type:	AWR1843BOOST mmWave radar from Texas Instruments.	Phoenix mmWave radar from Arbe Robotics.	IWR6843 60-64GHz mmWave radar from Texas Instruments.
Annotations:	Mesh annotated by VICON motion capture system and generated by SMPL.	55 keypoints are annotated by the OptiTrack Mocap system; Mesh is generated by Mosh++ and SMPL-X.	2D keypoints are obtained by HRNet-w48 from two-view infra-red cameras; 3D keypoints are calculated by triangulation.
Point format:	Cartesian: (x, y, z); PC attributes: (range, velocity, energy).	Cartesian: (x, y, z); PC attributes: (velocity, amplitude, energy).	Cartesian: (x, y, z); PC attributes: (velocity, intensity).
Public or not:	No.	Yes.	Yes.
# **of subjects:**	Not mentioned.	20 (10 males, 10 females).	40 (29 males and 11 females)
# **of actions:**	Not mentioned.	100 motions (16 static poses, 9 torso motions, 20 leg motions, 25 arm motions, 3 neck motions, 14 sports motions, 7 daily indoor motions, and 6 kitchen motions).	27 actions (14 daily activities and 13 rehabilitation exercises) for a duration of 30 seconds.
# **of frames:**	Not mentioned.	39892 frames for training and 28048 frames for testing.	133920 frames for training and 38400 frames for testing.
Scenes:	Normal and occlusion.	Lab1, Lab2, Furnished, Poor_lighting, Rain, Smoke, and Occlusion.	Normal, Cross-subject, and Cross-environment.

Model training. We follow DDPM [7] for faster convergence of the diffusion model. Firstly, diffusion step $k \in [1..K]$ and $\varepsilon \in \mathcal{N}(0, I)$ are randomly sampled, and H_k is calculated according to Eq. 3. Then, the diffusion model is trained to approximate $\hat{\varepsilon}$ to ε, rather than directly approximate $\hat{H}_{k-1}$ to H_{k-1}. The learning objective following DDPM is formulated as:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{k\sim[1,T]}\mathbb{E}_{\varepsilon\sim\mathcal{N}(0,I)}||\varepsilon - \hat{\varepsilon}_\theta(H_k, k, C)||_2^2. \tag{5}$$

3 Evaluated Datasets

As shown in Table 1, we present a comparison of existing datasets for point-cloud-based mmWave human pose estimation (HPE), focusing on several key attributes including mmWave radar sensor type, keypoint annotations, radar point cloud format, dataset size, and the variety of scenes covered in the dataset.

mmMesh [15]. mmMesh is the systematic work proposing mmWave human sensing (mesh reconstruction) with an off-the-shelf commercial mmWave sensor, AWR1843BOOST mmWave from Texas Instruments [13]. It provides a systematic way to annotate human mesh (including human keypoints) with the VICON system and SMPL [10] algorithm. The mmMesh dataset is presented for comparison, as it is not public and only contains normal and occlusion scenarios.

mmBody [2]. mmBody dataset is a public dataset for mesh reconstruction with multi-modal sensors: depth camera, RGB camera, and mmWave radar. Specifically,

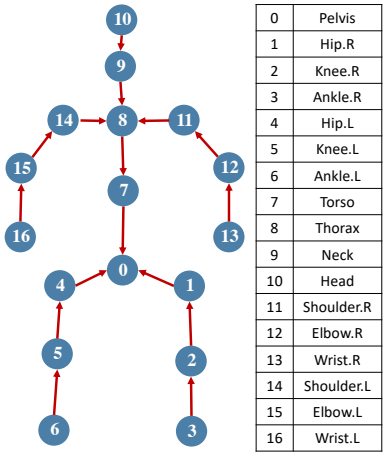
Phonix mmWave radar from Arbe Robotics is chosen as the mmWave sensor, which extracts thousands of radar points for scene detection. Still, the detected radar point clouds are noisy and sparse compared to RGB and depth sensors. The dataset contains daily-life motions, with heterogeneous human motions including motion of the torso, leg, arm, etc. Due to numerous subtle limb motions in the dataset, it is challenging to accurately predict human poses. To evaluate the robustness of mmWave HPE, the dataset includes various cross-domain scenes (lab2 and furnished) and adverse scenes (dark, rain, smoke, and occlusion). Meanwhile, except for lab2 containing seen subjects, all other scenes contain unseen subjects for testing, which is challenging for the model's generalizability. Our experiment is conducted following the mmBody settings [2].

mm-Fi [16]. mm-Fi offers a broader scope of human sensing, including action recognition and HPE, leveraging a variety of multi-modal sensors: RGB(D), LIDAR, WiFi, and mmWave radar. It is a large-scale dataset with 40 subjects participating and over 15k frames for training and testing. The mmWave radar utilized in the dataset is IWR6843 60-64GHz mmWave, which is a low-cost option generating a limited number of radar points. The point cloud format omits redundant range features. Meanwhile, different from the other two datasets, the model is trained in a self-supervised manner, as the human pose annotations are obtained by RGB image using HRNet-w48 [12]. The annotations are rather unstable compared to the motion capture systems. To test the model's generalizability, the dataset also proposes the cross-subject and cross-environment splits. Our experiment follows protocol 1 (P1) to include all daily-life activities and adopt all splitting methods.

4 Detailed Experiment Settings

4.1 Data Preprocessing

Radar point clouds preprocessing. Due to radar sparsity and the occasional miss-detection, we follow [1] to concatenate adjacent frames to enrich the number of points. Specifically, 4 frames are concatenated for mmBody and 5 frames are concatenated for mm-Fi. Further, since mmWave radar point clouds are generated by the targets with salient Doppler velocity, the number of radar points is frame-wise variant. As a result, to enable mini-batch training using PyTorch dataloader [11], we perform zero-padding (null points with 0 values) to guarantee the invariant input tensor shape. For mmBody, we zero-padding the point clouds to 5000 points, while a dynamic padding technique [16] is applied for mm-Fi. Moreover, to handle noisy radar points resulting from environmental reflection and interference, we perform point cloud cropping to select only the points within the region of human activities. For mmBody, the region of human activities is centered at the ground-truth pelvis location, with a region size of (x:$\pm$1.6m, y:$\pm$1.6m, z:$\pm$1.6m). However, for mm-Fi, as point clouds are generated solely by moving targets, cropping is unnecessary.



0	Pelvis
1	Hip.R
2	Knee.R
3	Ankle.R
4	Hip.L
5	Knee.L
6	Ankle.L
7	Torso
8	Thorax
9	Neck
10	Head
11	Shoulder.R
12	Elbow.R
13	Wrist.R
14	Shoulder.L
15	Elbow.L
16	Wrist.L

Fig. 2: Selected keypoints (ID and names) for mmWave human pose estimation. Red arrows indicate the select limbs.

Human pose and limb-length preprocessing. Our ground truth keypoints $H = \{h_1, .., h_{17}\}$ are selected according to Figure 2. Following [8] to construct ground-truth human poses, we perform pose normalization by pelvis alignment: subtracting the pelvis position h_1 from every keypoint $h_{1:17}$ of the skeleton. It's worth noting that the pelvis alignment is missing for the mm-Fi dataset, resulting in a higher Mean Per Joint Position Error (MPJPE). Consequently, we incorporate human pose normalization for a fair comparison. To calculate the ground truth 16 limb-length $L = \{l_1, .., l_{16}\}$, we compute the $\mathcal{L}_2$ distance between two adjacent human keypoints, as selected according to Figure 2.

4.2 Implementation of mmDiff

Conditional diffusion model. The GCN encoders, GCN blocks, and GCN decoders are designed following [18] and [6]. All MLPs are implemented as (LayerNorm, Linear, Dropout, ReLU, and Linear), and the temporal 1D-convolution extractor g_2^{tem} is implemented as (Conv1D(k=3), Dropout, ReLU, Conv1D(k=3), MaxPool). The conditional diffusion model adopts a Graph Convolution Network (GCN) architecture inspired by GraphFormer [18] as its backbone. The pose encoder and decoder utilize Chebyshev graph convolution layers [3] to project human poses to 96-dimensional pose embeddings. The GCN block is implemented by stacking a Chebyshev graph convolution layer and a graph attention layer, with skip connection. The GCN backbone consists of 5 GCN blocks. Chebyshev graph convolution layer first projects the 96-dimensional pose embeddings to 96-dimensional hidden feature embeddings. Subsequently, each graph attention layer performs self-attention with 4 attention heads in a 96-dimensional hidden feature space. During training, the GCN backbone applies a dropout rate of 0.25 and an exponential moving average (EMA) rate of 0.999 0.999. The condi-

tional embeddings are added at the start of each GCN block. Notably, as the Graph-Conditional Generation (GRC) model is trained before the diffusion training, the extracted conditional embeddings C^{glo} require an additional projection to align with the human pose feature embeddings. This alignment is achieved through a Chebyshev graph convolution layer projecting from 64 dimensions to 96 dimensions.

Radar point cloud encoder. Since mmBody and mm-Fi utilize different mmWave radars for sensing, and mm-Fi has sparser point clouds, the design of the radar point cloud (PC) encoder within the Global Radar Context (GRC) model differs between the two datasets. For mmBody, the PC encoder adopts the point 4D convolution following the P4Transformer benchmark [5]. Anchors are selected using farthest point sampling (FPS) with a spatial stride of 32, resulting in the selection of 312 anchors. Ball query is then applied to retrieve nearby radar points within a radius of 0.1 and 32 samples. A temporal convolution with a kernel size of 32 and a stride of 2 is subsequently applied to encode the nearby PCs into a 1024-dimensional PC feature space. On the other hand, for mm-Fi, the radar PC encoder is designed following the PointTransformer approach [17]. It utilizes 5 point-attention blocks with $n_{\text{neighbor}} = 16$ and $d_f = 128$ feature dimensions. After the radar PC feature encoders, the extracted PC features are concatenated with randomly initialized joint feature templates and fed into the global-transformer layers [14].

Global and local transformers. The global transformer consists of $D = 10$ transformer layers, each utilizing $H = 8$ attention heads and a hidden feature dimension of 256. The local lightweight transformer consists of $D = 5$ transformer layers, $H = 8$ attention heads, and a hidden feature dimension of 96. In each attention layer of both transformers, skip connections are applied. Additionally, after the local transformer, each anchor computes the proportion of points that are within a distance threshold of $thre = 0.04$m. This proportion serves as a measure of reliability for the local features. We incorporate it by multiplying the local features with the calculated proportion (which falls in the range $[0, 1]$). This approach helps to weigh the local features based on their distance to nearby points, enhancing the model's ability to focus on more relevant information.

4.3 Implementation of other Compared Methods

The methods using other modalities, e.g., RGB and depth are directly recorded from mmBody [2]. Other methods use mmWave radar point clouds (PCs) as the input modality, which are described in detail:

mmWave-based HPE methods. For a more direct comparison of performance gains achieved by mmDiff, it's important to note that the radar point cloud (PC) encoders of P4Transformer [5] and PointTransformer [17] are the same with mmDiff's radar PC encoder.

In the case of P4Transformer, which serves as the benchmark for mmBody, the PC encoder first utilizes a point4D convolution layer to extract PC features $F^r \in \mathbb{R}^{P\times 1024}$. Subsequently, a global transformer performs self-attention on these features. Unlike GRC modules, the input to the global transformer is solely the PC features F^r without any joint feature templates $\bar{F}^j$. Following the global transformer, a nonparametric max-pooling layer aggregates $\mathbb{R}^{P\times 1024}$ PC features into $\mathbb{R}^{1024}$. This aggregated feature is then dimensionally reduced into a $\mathbb{R}^{64}$ feature space through an MLP and decoded into $\mathbb{R}^{17*3}$ human poses. On the other hand, PointTransformer, which serves as the benchmark for mm-Fi, employs a point attention PC encoder to extract PC features. Similar to P4Transformer, a global transformer performs self-attention on these features. Then, a max-pooling layer aggregates PC features, followed by two MLPs for dimension reduction and pose decoding. Lastly, mmMesh [15] is implemented following its open-source design without modification.

Camera-based 3D HPE methods. Since camera-based 3D Human Pose Estimation (HPE) methods cannot directly utilize radar point clouds as input, we adopt a different approach. These methods typically involve pose-lifting, where 2D human poses are used as input to predict 3D human poses. Therefore, we fine-tune these models for 3D pose refinement: Given coarsely estimated 3D human poses $\tilde{H}$, the models aim to estimate cleaner 3D human poses $\hat{H}$ by eliminating noise within $\tilde{H}$.

To ensure a fair comparison, the input coarse human poses $\tilde{H}$ are the same for mmDiff and the compared 3D HPE methods. We implement PoseFormer [19] following its open-source design. Similarly, we implement DiffPose [6] using the same GCN diffusion backbone as used in mmDiff. This consistent setup allows for a direct comparison between the performance of mmDiff and these 3D HPE methods in refining 3D human poses.

Other methods In this section, we provide a discussion of other mmWave-based methods that do not utilize PCs as the input modality, i.e. using raw radar signals or Range-Doppler (RD) maps. Specifically, Hupr [9] and MI-Mesh [4] utilize raw radar signal to perform HPE. We summarize the difference between mmDiff with HuPR and MI-Mesh in Table 2, regarding input modalities, evaluation dataset pros and cons, preprocessing methods, and reported joint errors. Quantitative comparisons are limited by the absence of datasets containing both RF raw data and radar PCs.

5 Supplementary Results

5.1 Visualization of Diffusion Process

As shown in Figure 3, we provide the visualization of progressive noise elimination of diffusion-based human pose estimation, where mmDiff performs progressive noise elimination and generates human poses from coarse to fine.

Methods	Input Modalities	Evaluation Dataset	Data Preprocessing	Joint Error
HuPR	Raw signal reflections from two synchronized radars.	Raw data: (1) More information to process, thus computationally expensive. (2) Contain rich environmental information, thus subjective to interference and domain shift.	Tailored algorithms to obtain range-doppler-azimuth-elevation map.	MPJPE: 68 mm
MI-Mesh	Radar PCs and cameras fusion.	Camera: Not applicable to privacy-preserving scenarios where cameras cannot be deployed.	Techniques to calibrate phase and handle multi-path effect.	MPJPE: 69 mm
mmDiff	Noisy PCs from single radar.	Noisy radar PCs: (1) Privacy-preserving and (2) Better cross-environment capability.	Off-the-shelf PC generation algorithm.	MPJPE: 68 mm (mmBody) and 65 mm (mm-Fi)

Table 2: Qualitative Comparison with HuPR and MI-Mesh.

The pose is refined from yellow to green. We can observe progressively improved keypoint accuracy during the pose refinement process: For the lab1 scene, the hands' locations are approaching the head with the increment of diffusion steps; For the rain scene, the legs' locations are also corrected to better reflect the walking poses. Meanwhile, we observe the correction of pose deformity in the smoke and dark scenes, as the shoulders' locations are progressively refined to the ground truth location. Furthermore, due to the limited pages in the main paper, we provide the visualization of motion continuity consistency and limb-length distribution of the spine length here. As shown in Figure 4, we observe the correction of erroneous frames w/ temporal motion consistency.

5.2 Visualization of Global Radar Context

To demonstrate the effectiveness in extracting more robust joint-wise features, we perform further visualization of the transformer's attention heatmap using the global transformer in the Global Radar Context (GRC). First, as shown in Figure 5, the global transformer extracts the self-correlation within the PC features F^r and the inter-correlation between F^r and F^j. In Figure 6, we further demonstrate the feature extraction of different joints does not affect each other, as they focus on different parts of the PC features. Such quality facilitates joint-wise feature extraction. Finally, in Figure 7 and 8, we visualize the attention region of different joint features. We observe that the feature extraction of detected joints focuses on the correct-detected PC region, which facilitates more robust feature extraction. Though feature extraction of undetected joints focuses on the wrong part, it does not affect the feature extraction of other detected joints. Therefore, the joint-wise feature extraction is more robust for detected joints.

5.3 Visualization of Local Radar Context

To illustrate the effectiveness of dynamic local PC selection in the Local Radar Context (LRC), we provide a visualization of the local PC selection. We examine the local PC selection during model inference, as shown in Figure 9.

The dynamic joint anchors utilize intermediate diffusion poses $\hat{H}_k$, starting from coarsely estimated poses $\tilde{H}$ (as in $k = 25$) and progressively refined ($k = 25 \rightarrow 0$). Though initially, the joint anchor from $\hat{H}_k$ failed to select the upper left local PCs around the human neck, the anchor is progressively refined to the ground truth location. Finally, as $k = 0$, the upper left local PCs are considered for more robust local feature extraction. On the other hand, static joint anchors only incorporate $\tilde{H}$, leading to biased local PC selection.

5.4 Effect of Hyper-parameters

Table 3: Parameter sensitivity analysis of the diffusion steps K. We record joint errors by MPJPE in white and PA-MPJPE in gray. Bold is the best and red is the worst.

K	Basic Scenes						Adverse Environment								Average	
	Lab1		Lab2		Furnished		Rain		Smoke		Poor_lighting		Occlusion			
12	61.3	50.23	71.37	66.08	70.52	51.76	76.41	60.43	78.75	63.86	67.84	49.67	71.22	51.89	71.06	56.27
25	59.52	**47.85**	69.36	**61.23**	67.15	**49.19**	**71.03**	58.40	76.92	**62.25**	65.08	**47.47**	**67.47**	49.54	**68.08**	**53.71**
36	**59.21**	48.68	69	63.42	67.75	49.79	72.38	58.95	**76.41**	62.41	65.54	48.37	69.09	51.12	68.48	54.68
50	60.16	48.83	**66.51**	61.39	67.8	50.18	73.44	**57.18**	80.65	63.04	65.38	48.11	68.11	**49.46**	68.86	54.03
60	59.97	48.65	67.61	62.53	**67.07**	49.32	73.12	57.37	80.02	62.73	**65.01**	47.85	68.34	49.62	68.73	54.01

Table 4: Parameter sensitivity analysis of the β scheduling for the diffusion model. We record joint errors by MPJPE in white and PA-MPJPE in gray. C denotes constant β scheduling, L denotes linear β scheduling. Bold is the best and red is the worst.

β scheduling	Basic Scenes						Adverse Environment								Average	
	Lab1		Lab2		Furnished		Rain		Smoke		Poor_lighting		Occlusion			
C: 0.001	**59.37**	48.66	67.24	60.69	65.83	48.65	72.24	57.77	79.82	63.55	64.34	47.1	64.81	48.57	**67.66**	**53.57**
C: 0.002	61.29	49.37	72.98	61.7	66.87	48.43	70.86	57.77	77.47	63.74	63.72	46.76	71.57	51.45	69.25	54.17
L: [0.0001, 0.001]	59.52	**47.85**	69.36	61.23	67.15	49.19	71.03	58.40	76.92	62.25	65.08	47.47	67.47	49.54	68.08	53.71
L: [0.0001, 0.002]	60.87	48.39	66.79	61.74	66.66	48.63	74.92	57.88	80.62	62.68	65.26	47.61	69.45	49.96	69.22	53.84

Table 5: Parameter sensitivity analysis of the limb loss weighting parameter λ for the structural limb-length consistency module. We record joint errors by MPJPE in white and PA-MPJPE in gray. Bold is the best and red is the worst.

λ	Basic Scenes						Adverse Environment								Average	
	Lab1		Lab2		Furnished		Rain		Smoke		Poor_lighting		Occlusion			
2	59.35	47.99	**67.71**	61.97	**66.99**	49.63	73.04	57.85	78.77	**62.06**	**64.99**	47.80	68.29	49.70	68.45	53.86
5	58.65	**47.66**	68.91	62.25	67.86	50.33	71.42	**57.58**	77.19	62.84	65.50	47.86	67.79	**49.15**	68.19	53.95
8	59.47	48.47	68.07	61.90	67.83	49.69	72.34	57.75	78.58	62.72	65.59	48.26	68.68	51.25	68.65	54.29
10	59.52	47.85	69.36	**61.23**	67.15	**49.19**	**71.03**	58.40	**76.92**	62.25	65.08	**47.47**	67.47	49.54	68.08	**53.71**
15	**58.36**	47.75	68.01	62.65	67.80	49.92	71.16	57.87	78.22	63.37	65.49	47.94	**67.00**	49.66	**68.00**	54.17

Table 6: Parameter sensitivity analysis of the historical pose sequence length Δt for the temporal motion consistency module. We record joint errors by MPJPE in white and PA-MPJPE in gray. Bold is the best and red is the worst.

Δt	Basic Scenes						Adverse Environment								Average	
	Lab1		Lab2		Furnished		Rain		Smoke		Poor_lighting		Occlusion			
2	60.86	48.54	69.32	62.82	67.72	49.68	72.94	57.85	81.39	63.72	65.19	48.18	69.25	50.99	69.52	54.54
4	59.59	47.77	68.63	62.75	67.11	49.45	73.05	57.59	79.91	63.12	65.50	47.93	68.70	49.80	68.93	54.06
6	59.52	47.85	69.36	**61.23**	67.15	49.19	**71.03**	58.40	**76.92**	62.25	65.08	47.47	67.47	49.54	**68.08**	53.71
8	**58.44**	47.89	68.79	64.02	68.00	50.44	72.15	58.52	77.18	**61.81**	65.86	48.58	**67.12**	50.19	68.22	54.49
10	58.57	**47.51**	**68.05**	62.59	**66.96**	**49.01**	73.10	**57.47**	78.14	62.46	**64.90**	**47.34**	67.74	**49.23**	68.21	**53.66**

Effect of diffusion steps K. As shown in Table 3, we present how the joint error is affected by the number of diffusion steps K on mmBody [2]. When diffusion steps $K < 25$, we observe that the increment of K can improve the performance of mmDiff. As with more iterations, mmDiff potentially can handle noisier human poses. However, with the number of diffusion steps $K > 25$, the performance of mmDiff converges. We argue that the pose noise within the coarsely estimated human poses is well handled with 25 diffusion steps.

Effect of β scheduling. As shown in Table 4, we present how the β scheduling affects the diffusion-based HPE. We observe different β scheduling significantly affects the mmDiff performance, as the selected noise scale should accurately approximate the noise contained by the coarsely estimated human poses. For linear scheduling, if the β range is increased (as in range$[0.0001, 0.002]$), the performance drops significantly as the reverse diffusion process is not stable. We also observe that constant β scheduling also performs well, as long as the noise scale is selected properly.

Effect of the weighting parameter λ for $\mathcal{L}_{\text{diff}}$. As shown in Table 5, we present how the model performs with different weighting parameters λ to integrate the limb loss $\left|L - \hat{L}\right|$ into the diffusion loss $\mathcal{L}_{\text{diff}}$. We observe that our model is not sensitive to the selection of λ, with 68.27 ± 0.24 (MPJPE) statistically lower than the performance without spatial limb consistency, 69.16 (MPJPE). We argue that as long as the limb loss is presented, the model can estimate a subject's limb length with acceptable accuracy.

Effect of the sequence length Δt. As shown in Table 6, we present how our approach is affected by the number of adjacent time frames used in the temporal motion consistency module. When the sequence length $\Delta t \leq 6$, we observe that the increment of Δt can improve the performance. As with longer sequence lengths, mmDiff potentially can extract more reliable and robust motion patterns of the subjects. However, as the sequence lengths $K > 6$ keep increasing, the performance of mmDiff begins to drop, especially in adverse environments. As the training of mmBody is conducted on basic scenes, the increment of sequence

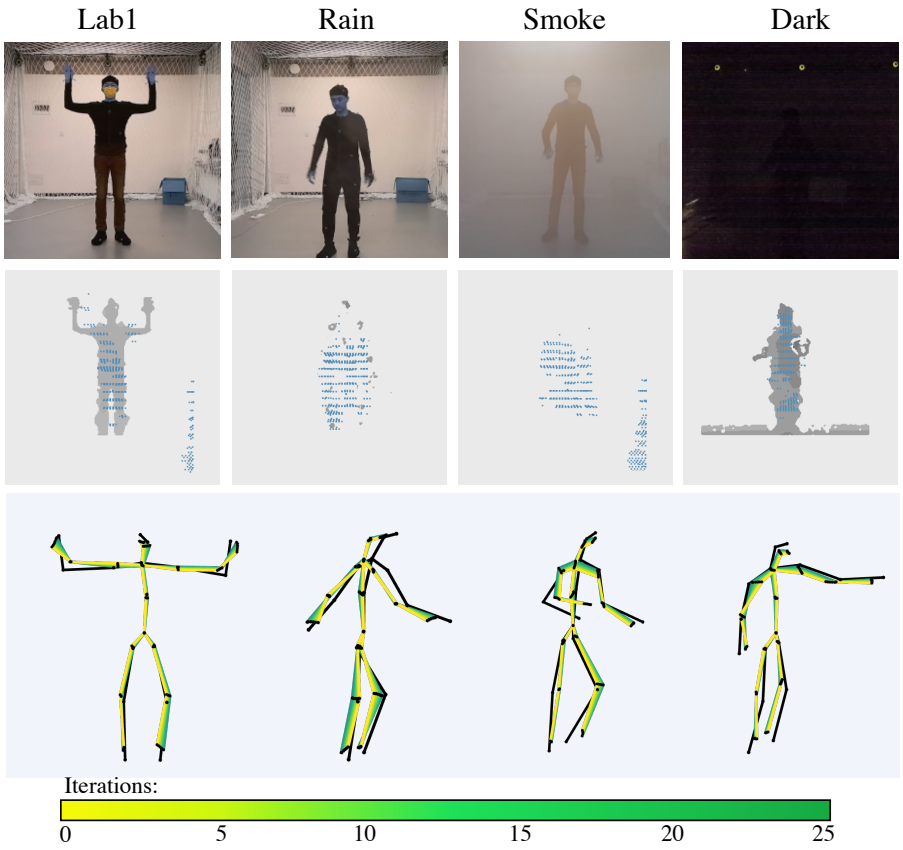


Fig. 3: Visualization of diffusion-based human pose estimation with diffusion 25 steps. We use gradient colors (from yellow to green) to illustrate the refined poses of different iterations. The yellow pose is the initialized coarse human pose, and the green pose is the final refined pose.

length tends to overfit the model to basic scenes, causing a performance drop in adverse environments. As a result, $\Delta t = 6$ is the optimum for the mmBody dataset.

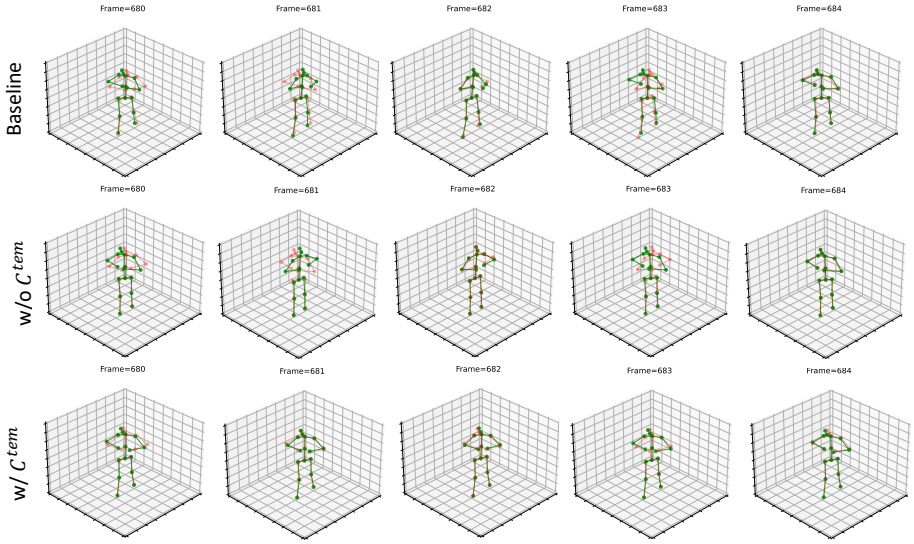


Fig. 4: Visualization of motion continuity on mm-Fi, with PointTransformer baseline [17], ours w/o C^{tem} and ours w/ C^{tem}. Green poses are the ground truth and red poses are the prediction. We can observe occasion erroneous frames are corrected based on smooth motion patterns.

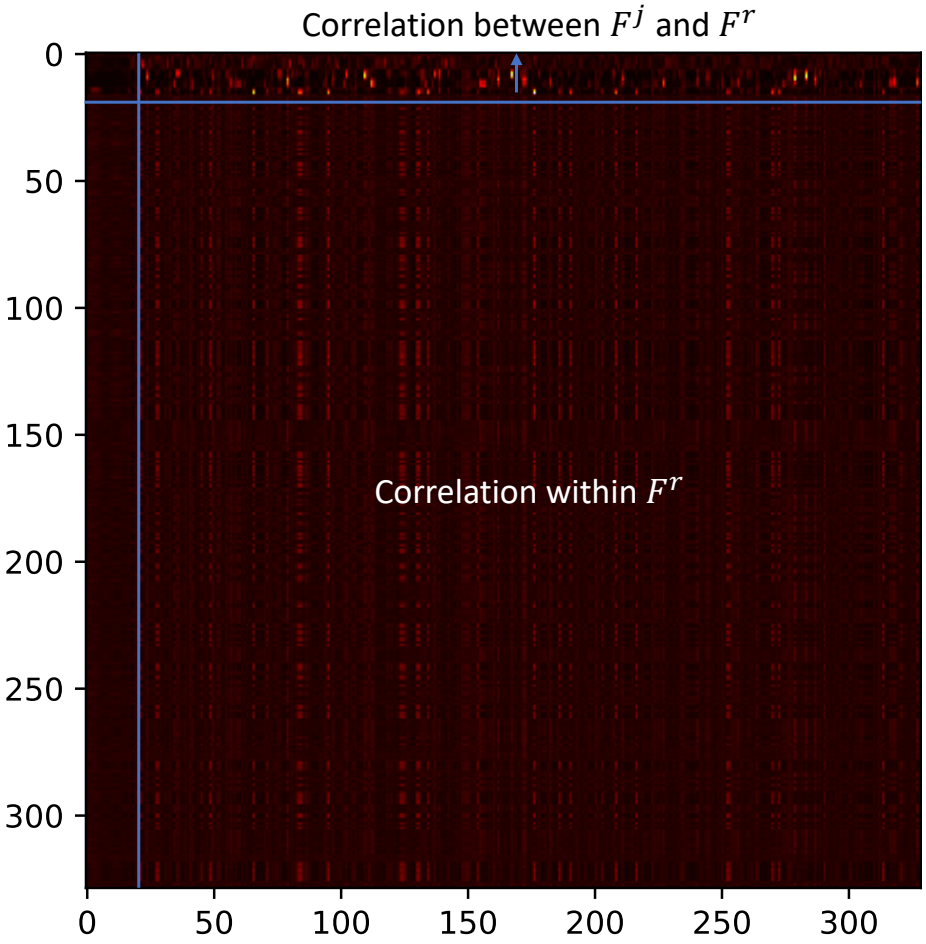


Fig. 5: Visualization of the correlation using the global transformer ϕ^g of GRC tested on mmBody: within PC features F^r, and between joint features F^j and PC features F^r.

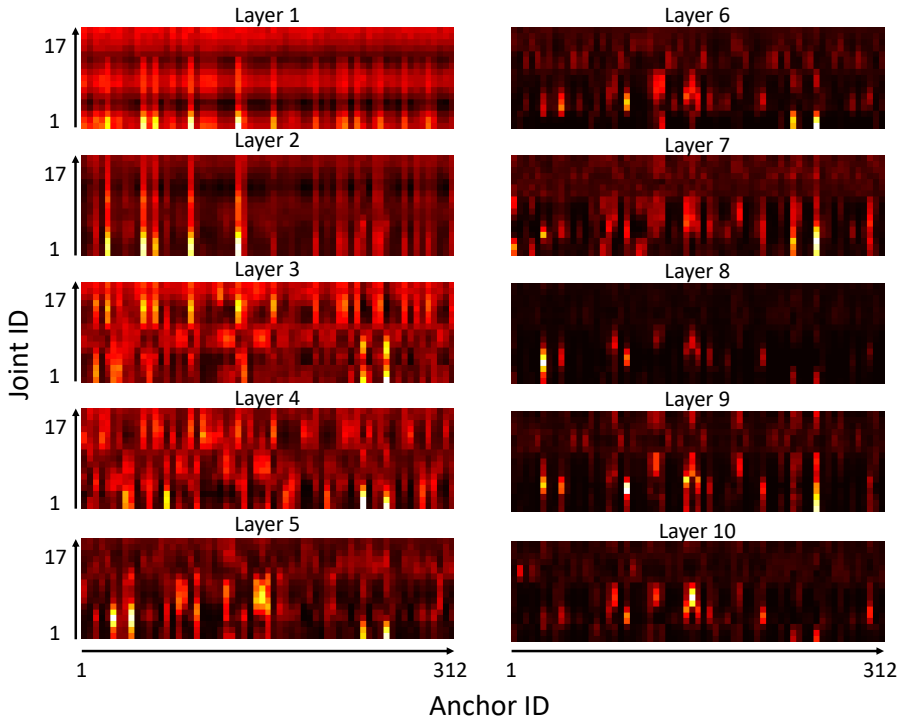


Fig. 6: Visualization of the correlation between joint features F^j and PC features F^r, using the global transformer ϕ^g of GRC tested on mmBody. Feature extraction of different joints depends on individual correlation with the PC feature, which is less affected by other joint features after 5 transformer layers.

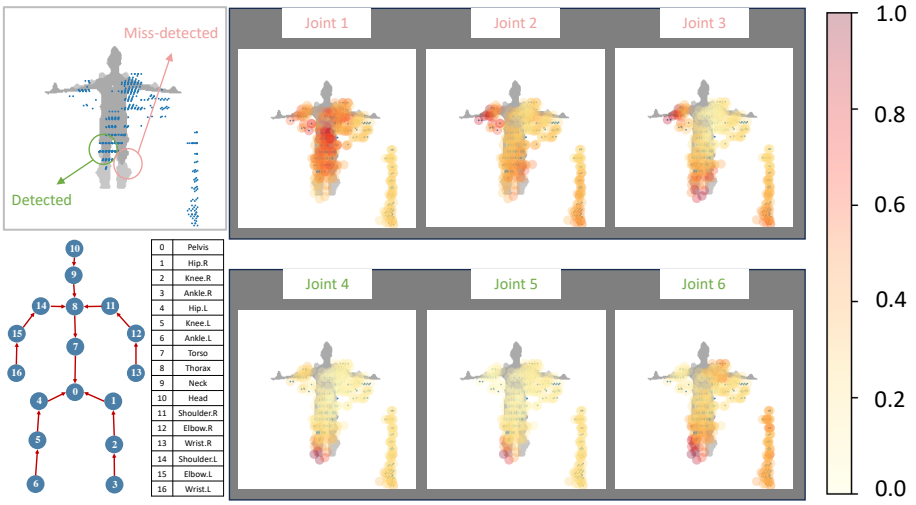


Fig. 7: Visualization of the attention when performing feature extraction of joints 1-6 (legs). Each joint performs individual feature extraction. For detected joints 4-6(left leg), the attention is more concentrated as correctly focuses on the left leg part, extracting more robust features. The feature extraction is more distracted and less reliable for undetected joints 1-3 (right leg) due to miss-detection.

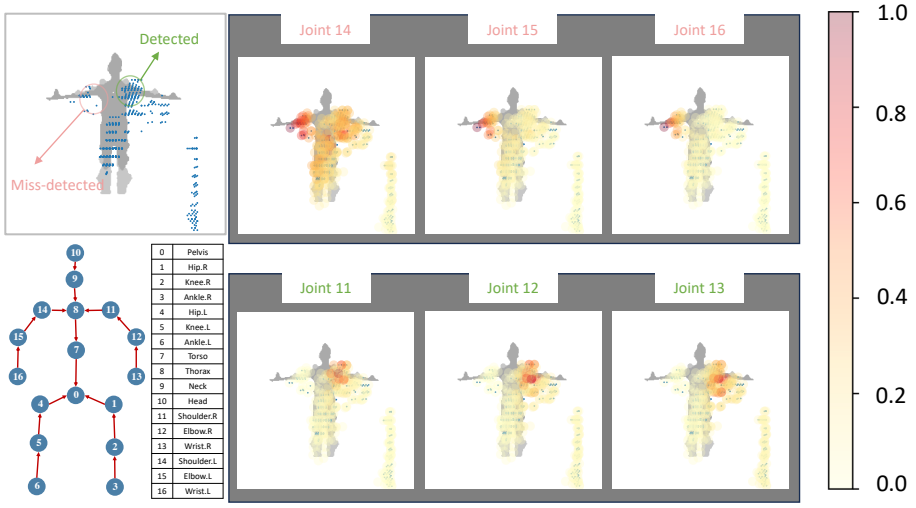


Fig. 8: Visualization of the attention when performing feature extraction of joints 11-16 (arms). As reflected by the attention heatmap, detected joints 11-13(right arm) have more robust features, preventing the influence of undetected joints 14-16 (left arm).

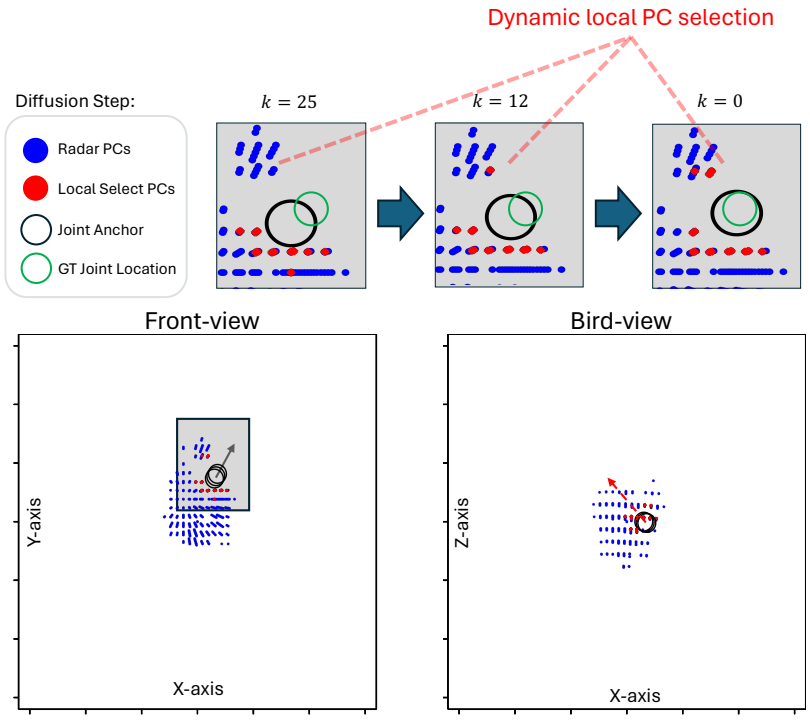


Fig. 9: Visualization of the local PC selection using dynamic joint anchors (right shoulder) during model inference phase. Initially, as $k = 25$, the joint anchor is derived from the coarsely estimated pose $\tilde{H}$, failing to select the upper left local PCs. The dynamic joint-anchor starts from $\tilde{H}$ and is progressively refined with the diffusion steps $k = 25 \rightarrow 0$. Finally, as $k = 0$, the upper left local PCs are selected for more robust local PC features. As the upper left PCs correspond to the human neck, a more robust local feature is extracted considering both shoulder and neck PCs.

References

1. An, S., Ogras, U.Y.: Fast and scalable human pose estimation using mmwave point cloud. In: Proceedings of the 59th ACM/IEEE Design Automation Conference. pp. 889–894 (2022)
2. Chen, A., Wang, X., Zhu, S., Li, Y., Chen, J., Ye, Q.: mmbody benchmark: 3d body reconstruction dataset and analysis for millimeter wave radar. In: Proceedings of the 30th ACM International Conference on Multimedia. pp. 3501–3510 (2022)
3. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. Advances in neural information processing systems **29** (2016)
4. Ding, H., Chen, Z., Zhao, C., Wang, F., Wang, G., Xi, W., Zhao, J.: Mi-mesh: 3d human mesh construction by fusing image and millimeter wave. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies **7**(1), 1–24 (2023)
5. Fan, H., Yang, Y., Kankanhalli, M.: Point 4d transformer networks for spatio-temporal modeling in point cloud videos. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 14204–14213 (2021)
6. Gong, J., Foo, L.G., Fan, Z., Ke, Q., Rahmani, H., Liu, J.: Diffpose: Toward more reliable 3d pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13041–13051 (2023)
7. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)
8. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. IEEE transactions on pattern analysis and machine intelligence **36**(7), 1325–1339 (2013)
9. Lee, S.P., Kini, N.P., Peng, W.H., Ma, C.W., Hwang, J.N.: Hupr: A benchmark for human pose estimation using millimeter wave radar. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 5715–5724 (2023)
10. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: Smpl: A skinned multi-person linear model. In: Seminal Graphics Papers: Pushing the Boundaries, Volume 2, pp. 851–866. ACM (2023)
11. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019)
12. Prabhakara, A., Jin, T., Das, A., Bhatt, G., Kumari, L., Soltanaghai, E., Bilmes, J., Kumar, S., Rowe, A.: High resolution point clouds from mmwave radar. In: 2023 IEEE International Conference on Robotics and Automation (ICRA). pp. 4135–4142. IEEE (2023)
13. Rao, S.: Introduction to mmwave sensing: Fmcw radars. Texas Instruments (TI) mmWave Training Series pp. 1–11 (2017)
14. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
15. Xue, H., Ju, Y., Miao, C., Wang, Y., Wang, S., Zhang, A., Su, L.: mmmesh: Towards 3d real-time dynamic human mesh construction using millimeter-wave. In: Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services. pp. 269–282 (2021)

16. Yang, J., Huang, H., Zhou, Y., Chen, X., Xu, Y., Yuan, S., Zou, H., Lu, C.X., Xie, L.: Mm-fi: Multi-modal non-intrusive 4d human dataset for versatile wireless sensing. arXiv preprint arXiv:2305.10345 (2023)
17. Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 16259–16268 (2021)
18. Zhao, W., Wang, W., Tian, Y.: Graformer: Graph-oriented transformer for 3d pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20438–20447 (2022)
19. Zheng, C., Zhu, S., Mendieta, M., Yang, T., Chen, C., Ding, Z.: 3d human pose estimation with spatial and temporal transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11656–11665 (2021)