

LiDAR-HMR: 3D Human Mesh Recovery From LiDAR

Bohao Fan , Wenzhao Zheng , *Member, IEEE*, Jianjiang Feng , *Member, IEEE*, and Jie Zhou , *Fellow, IEEE*

Abstract—Human mesh recovery (HMR) holds significant utility in many applications. Studying HMR involving various types of sensors is necessary, as it enables the acquisition of human meshes in diverse scenes. Unlike HMR based on RGB images, HMR based on LiDAR has received considerably less attention in previous works. The major challenge in estimating human poses and meshes from sparse point clouds lies in the sparsity, noise, and incompleteness of LiDAR point clouds. To address these challenges, we propose a LiDAR-based 3D human mesh recovery algorithm, called LiDAR-HMR. This algorithm involves estimating a sparse representation of a human (3D human pose) and gradually reconstructing the body mesh. To better leverage the 3D structural information of point clouds, we propose a point-cloud-to-SMPL pipeline that uses the original point cloud features to guide the reconstruction. The experimental results on four publicly available datasets demonstrate the effectiveness of LiDAR-HMR.

Index Terms—3D human pose estimation, human mesh recovery (HMR), point cloud processing.

I. INTRODUCTION

3D HUMAN pose estimation (HPE) and human mesh recovery (HMR) in unconstrained scenes have long been goals in computer vision. Compared with HPE, HMR contains extra information on the shape and details of the human body, which extends to a wider range of applications such as human-computer interaction and identity recognition. Mainstream 3D HMR or HPE methods focus primarily on RGB inputs [3], [13], [17], [25], [26], [27], [29], [41], [46], [52] or multimodal fusion [2], [4], [39], [47], [51], with sparse research dedicated to point cloud input. In particular, LiDAR point clouds are often utilized as auxiliaries for RGB images in multimodal settings. However, compared with RGB images, point clouds are more robust to illumination conditions and offer better privacy protection, thereby deserving more research. With the introduction of several point cloud-based datasets [5], [7], [36] and 3D HPE or HMR algorithms such as LPFormer [44] and LiDARCap [24],

the feasibility of using LiDAR point clouds alone to perceive human meshes has been demonstrated.

Existing 3D HMR methods based on single-frame point clouds [16], [28] focus mainly on dense point clouds derived from depth maps. When captured by an RGB-D scanner in indoor settings, these point clouds are typically complete and dense, whereas the point clouds captured by LiDAR in outdoor scenes are sparse. Other methods [24], [38] utilize sequential LiDAR point clouds for HMR. However, these methods must process a complete point cloud sequence, which means that the algorithm cannot output human meshes in real time, and cannot be trained using point clouds with only single frame labels, such as the Waymo dataset [36]. Here, we consider a more practical and important setting: 3D HMR from single-frame LiDAR point clouds. This setting is analogous to image-based HMR, which is the basis of video-based HMR. This setting is quite challenging, as illustrated in Fig. 1, single-frame LiDAR point clouds have distinct characteristics: they are usually sparse, incomplete, self-occluding, and sometimes very noisy. As illustrated in Fig. 2, most point cloud-based HMR methods [16], [24], [25], [28] rely on neural networks to extract global features and regress SMPL parameters. However, the mapping from SMPL parameters to mesh vertices is inherently nonlinear. Direct regression of SMPL parameters from global features thus loses the correspondence between the point cloud and the human mesh vertices, which leads to a loss of information and diminishes its ability to fit data. This results in unsatisfactory HMR performance, especially under sparse point clouds. Specifically, [38] employed a framework of coarse-to-fine reconstruction and regress mesh vertices, which are less affected by the above. They acquire a global feature representation from the input data, incrementally adding local details to reconstruct the human mesh. However, global feature extraction often leads to a loss of local details, and subsequent coarse-to-fine reconstruction relies on the human mesh priors to fill in the missing vertices, rather than referring to local surface details from the input point cloud, which results in the loss of specific local details. In addition, their outputs do not include explicit human pose semantics, but only mesh vertex coordinates, which limits their potential application.

To effectively reconstruct human poses from sparse point clouds, we aim to more comprehensively use the local details in the input point clouds. Consequently, we developed a point-cloud-to-SMPL dense reconstruction pipeline, which follows a sparse-to-dense reconstruction process. We use a pose regression network (PRN), which is a point cloud-based analysis network to extract template human 3D poses, perform

Received 15 August 2024; revised 17 November 2024 and 31 December 2024; accepted 18 January 2025. Date of publication 22 July 2025; date of current version 21 October 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62321005 and Grant 62376132, and in part by the National Key Research and Development Program of China under Grant 2018AAA0102803. The associate editor coordinating the review of this article and approving it for publication was Dr. Li Cheng. (*Corresponding author: Jianjiang Feng.*)

The authors are with the Department of Automation, Tsinghua University, Beijing 100190, China (e-mail: fbh19@mails.tsinghua.edu.cn; wenzhao.zheng@outlook.com; jfeng@tsinghua.edu.cn; jzhou@tsinghua.edu.cn).

The codes are available at <https://github.com/soullessrobot/LiDAR-HMR>.

Digital Object Identifier 10.1109/TMM.2025.3590928

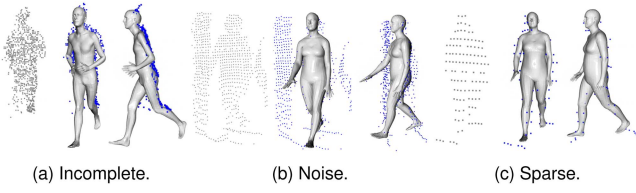


Fig. 1. Three challenges in 3D human mesh reconstruction from single frame sparse LiDAR point cloud. (Point clouds together with corresponding ground truth meshes (front view and side view).).

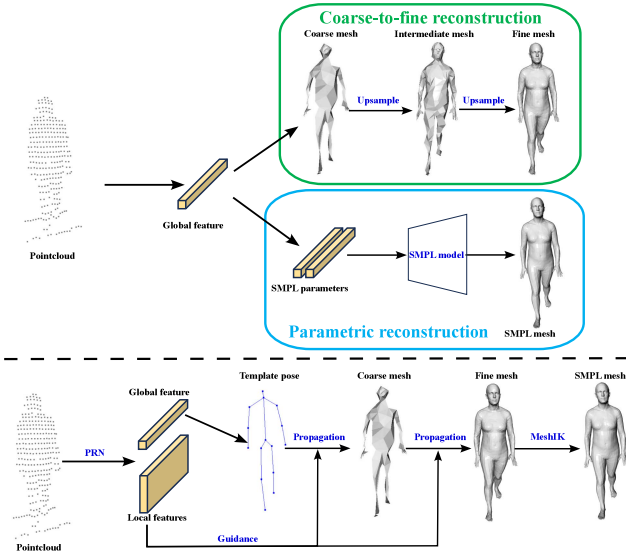


Fig. 2. Top: Most 3D HMR methods from point cloud inputs utilize the sparse global feature extracted from the initial point cloud, which will cause information loss in local details. Down: The proposed point-cloud-to-SMPL pipeline for 3D HMR. Point cloud features are utilized to guide the coarse-to-fine process of human meshes to obtain better local details, and point features from coarse meshes are propagated rather than upsampled to obtain features in the fine meshes. Finally, a differentiable module, named MeshIK, can obtain SMPL parameters from fine meshes.

pointwise segmentation and extract corresponding point cloud features. The extracted pointwise features can be directly utilized for mesh reconstruction in the subsequent mesh reconstruction network (MRN), thus constructing an end-to-end trainable network that combines point cloud analysis, pose estimation, and mesh reconstruction. Specifically, features between meshes of different resolutions are propagated on the basis of their relationships rather than being upsampled. Finally, we propose a differentiable MeshIK module to resolve the SMPL pose and shape parameters from the estimated 3D human mesh. The whole network is end-to-end trainable and we utilize end-to-end multitask (HPE and HMR) training for the entire network. As shown in Fig. 3, our LiDAR-HMR can handle various types of outdoor scenes. Compared with the RGB-based methods, it is not affected by low illumination, which can be an important advantage in certain applications. The experimental results on four public datasets demonstrate that LiDAR-HMR not only achieves the best performance in HMR but also achieves superior performance in HPE.

In summary, the contributions of this paper are as follows:

- 1) We propose a novel point cloud-based pipeline for human mesh recovery, which effectively incorporates local details

from the point cloud into the reconstruction process to maximize the retention of local features.

- 2) We design a resolution-consistent feature propagation module that effectively inherits and constrains the feature relationships between meshes of varying resolutions throughout the reconstruction process from sparse to dense.
- 3) We emphasize and quantify the prevalent issue of local detail collapse in nonparametric mesh reconstruction methods, and we introduce a new module, MeshIK, to address this problem.
- 4) The proposed network, LiDAR-HMR, achieves state-of-the-art performance on several publicly available datasets.

II. RELATED WORK

In recent years, human pose estimation (HPE) and human mesh recovery (HMR) have emerged as a prominent research topics, encompassing various areas of study, including 2D HPE [1], [43], 3D HPE [11], [31], [32], [42], [44], [47], [50], and human mesh recovery (HMR) [18], [21], [27], [34]. In this section, we focus on 3D HPE and HMR methods based on point cloud inputs. Given the relatively small number of 3D HMR works that focus on point cloud inputs, we also provide an overview of 3D HMR algorithms based on RGB images for reference in mesh recovery techniques.

3D HPE from point clouds: 3D HPE from depth images is a long-standing research topic [8], [9], [12], [20], [31], [35], [45]. However, depth image-based methods can only be used in indoor scenes. Recently, with the proposal of 3D human pose datasets for point cloud scenes [5], [24], [36], several 3D pose estimation algorithms that are based on LiDAR point clouds have emerged. Zheng et al. [51] first proposed a multimodal 3D HPE network to fuse RGB images and point clouds, using 2D labels as weak supervision for 3D pose estimation. In follow-up work, Weng et al. [39] used simulation data to train a transformer-based pose estimation network and then proposed a symmetry loss to fine-tune with the actual LiDAR point cloud input. Ye et al. [44] proposed a multitask structure and used the object detection task as pretraining. They used a transformer as a point cloud encoder to regress human poses and achieved state-of-the-art effects on the Waymo [36] dataset. These methods have successfully demonstrated the feasibility of estimating human body information from sparse LiDAR point clouds.

3D HMR from point clouds: Contemporary HMR algorithms for point clouds are mainly based on dense point clouds [16], [28] or sequential point clouds [24], [38]. Specifically, Jiang et al. [16] utilized a PointNet-based network to obtain point cloud features and used them to regress SMPL pose and shape parameters. Liu et al. [28] employed a voting mechanism to obtain sparse features of the human skeleton, and then used them to regress SMPL pose and shape parameters. Using point cloud video, Li et al. [24] proposed a GRU-based pipeline for sequential point cloud processing, and obtained sequential SMPL parameters via a spatial-temporal graph convolution network (ST-GCN). These algorithms fail to effectively synthesize the local details in the point clouds and are less effective for



Fig. 3. Example results of LiDAR-HMR on multiperson scenes in the Waymo [36] dataset. RGB images are not utilized in the algorithm but they are illustrated for better visualization. LiDAR-HMR can reconstruct accurate human meshes under different illumination conditions.

sparse point cloud inputs. Specifically, Wang et al. [38] proposed a coarse-to-fine mechanism and reconstructed sequential human meshes via spatial-temporal mesh attention convolution. The reconstruction process mainly depends on the continuity of the sequence and the priori of the human body surface. Other works have attempted to address specific problems inherent in point clouds. Jang et al. [15] proposed a conditional variational autoencoder-based network to learn effective features from partial point clouds, with the simulated complete point clouds as guidance. And Zuo et al. [53] proposed a self-supervised HMR algorithm based on a Gaussian mixture model, which learns an effective conditioned posterior probability to remove noisy points. These methods are deeply based on simulation data; in real-world scenarios, it is difficult to obtain a sufficiently dense point cloud for full learning and modeling.

3D HMR from RGB images: 3D HMR algorithms from RGB images can generally be divided into parameterized and nonparameterized methods. Most previous works [18], [22], [25], [37] used parameterized human body models, such as SMPL [30], and focused on regressing parameters instead of the mesh itself. Given the pose and shape coefficients, SMPL is stable and practical for creating human meshes. However, as discussed in these papers [3], [23], [33], [48], estimating the parameters directly from features is not easy because of the nonlinear mapping from the input data to the target parameters. Instead of regressing parametric coefficients, nonparametric methods [3], [26], [27] directly regress vertices from input data. In previous studies, graph convolutional neural networks (GCNNs) [3], [23] were a popular option because they are able to model local interactions between adjacent vertices. However, they are less effective at capturing global features between vertices and body joints. To overcome this limitation, METRO [26] proposed a set of transformer-based architectures to model the global characteristics of vertices. However, compared with GNN-based methods [3], [23], it is less convenient for modeling local interactions. In subsequent work, mesh graphormer [27] used a combination of graph convolution and a transformer to obtain better results.

In previous HMR methods based on coarse-to-fine mechanisms, sparse mesh representations are obtained from global

features and lack sufficient 3D representation. During the reconstruction and fine-tuning process, the original features from the input data are not well encoded in the human body mesh reconstruction process. Specifically, the point cloud contains sufficient 3D occupancy clues and reliable human body surface details. We input the point cloud into the mesh reconstruction process to guide the local details of intermediate meshes.

III. METHOD

The proposed LiDAR-HMR network can be divided into three parts. As illustrated in Fig. 4, the PRN uses Point Transformer-v2 [40] as a feature extractor to encode the point clouds, and it decodes the point features to reconstruct a template human pose. Given the template pose and per-point features, the MRN utilizes the point clouds to reconstruct the fine human mesh progressively. Finally, we propose MeshIK to obtain the SMPL pose and shape parameters from the estimated 3D mesh vertices.

A. Pose Regression Network

Probabilistic modeling: With the input point cloud $\hat{P}$ of n points $(p_0, p_1, \dots, p_{n-1})$, we model the problem of estimating 3D pose J with m key points $(j_0, \dots, j_{m-1})$ from point cloud $\hat{P}$ as follows:

$$\max_J P(J|p_0, p_2, \dots, p_{n-1}). \quad (1)$$

Without the connection relationship between key points, we have:

$$P(J|p_0, p_2, \dots, p_{n-1}) = \prod_{k=0}^{m-1} P(j_k|p_0, p_2, \dots, p_{n-1}). \quad (2)$$

With the hypothesis of the normal distribution and independent isodistribution, the probability distribution of j_k can be

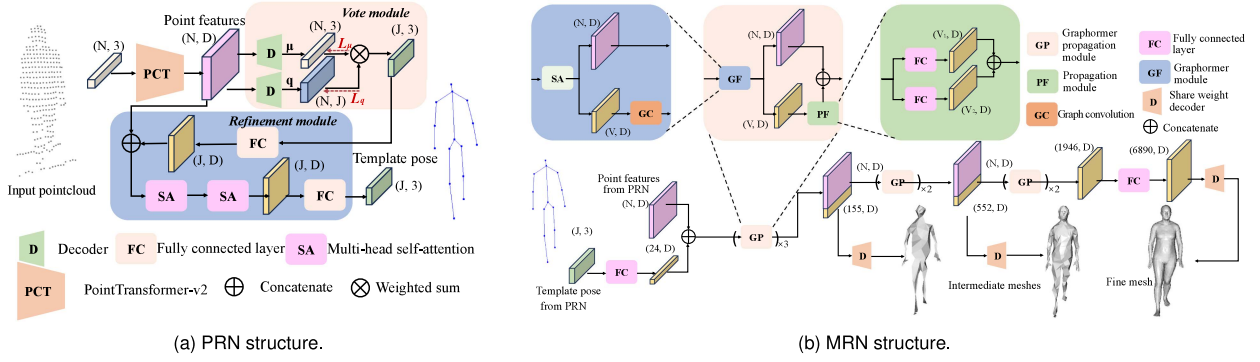


Fig. 4. The overall structure of the proposed modules. (a) For the pose regression network (PRN), the input point clouds are encoded by PointTransformer-v2 and decoded into q and μ to obtain a predicted human pose. The predicted pose is subsequently fed into two self-attention layers for refinement and completion. The red arrows indicate the loss functions for q and μ . The shape of the intermediate features is denoted as (x, y) . Specifically, N denotes the number of input points, J denotes the number of key points, and D denotes the fixed feature dimension for attention. (b) The MRN receives the point cloud features, which estimates a template pose from the PRN and gradually reconstructs the complete human mesh. We utilize a point cloud-based graphormer [27] for each intermediate resolution to introduce point cloud features during the reconstruction. Vertex features are inherited with a propagation module to model the parent-children relationship during the coarsening process. Finally, a fully connected layer is utilized to obtain the fine human mesh. The shape of the intermediate features is denoted as (x, y) . Specifically, N denotes the number of input points, V denotes the number of vertices, and D denotes the fixed feature dimension for attention.

described as follows:

$$P(j_k | p_0, p_2, \dots, p_{n-1}) = \prod_{i=0}^{n-1} P(j_k | p_i) = \prod_{i=0}^{n-1} \frac{1}{\sqrt{2\pi}\sigma_{ik}} e^{-\frac{(j_k - \mu_{ik})^2}{\sigma_{ik}^2}}, \quad (3)$$

where:

$$\mu_{ik} = p_i + d_{ik}, p_i \in R_3, d_{ik} \in R_3. \quad (4)$$

we assume that $d_i = d_{i0} = d_{i1} = \dots = d_{im-1}$ is the unified pointwise offset, and that μ_{ik} and σ_{ik} are parameters. We use a Gaussian distribution to approximate the joint posterior distribution as follows:

$$P(j_k | p_0, p_2, \dots, p_{n-1}) = e^C \prod_{i=0}^{n-1} e^{-\frac{(j_k - \mu_{ik})^2}{\sigma_{ik}^2}} = e^C e^{-\frac{(j_k - \hat{\mu}_k)^2}{\hat{\sigma}_k^2}}, \quad (5)$$

$$\frac{(j_k - \hat{\mu}_k)^2}{\hat{\sigma}_k^2} = -\sum_{i=0}^{n-1} \frac{(j_k - \mu_{ik})^2}{\sigma_{ik}^2}, \quad (6)$$

$$C = -\frac{n}{2} \ln(2\pi) - \sum_{i=0}^{n-1} \ln(\sigma_{ik}). \quad (7)$$

Letting $q_{ik} = \frac{1}{\sigma_{ik}^2}$, we have:

$$\hat{\mu}_k = \frac{\sum_{i=0}^{n-1} q_{ik} \mu_{ik}}{\sum_{i=0}^{n-1} q_{ik}}. \quad (8)$$

Letting $\sigma_{ik} \geq 1$ we obtain $q_{ik} \in (0, 1]$, which describes the confidence of p_i to vote for the position of keypoint j_k . The estimated model parameter $\hat{\mu} = (\hat{\mu}_0, \hat{\mu}_1, \dots, \hat{\mu}_{m-1})$ denotes the estimated template pose J_0 with the maximized probability.

Estimate and complete human pose: The overall structure of the PRN is shown in Fig. 4(a). We utilize PointTransformer-v2 [40] as a per-point feature extractor and two decoders to regress μ_{ik} and q_{ik} . This process is similar to the “vote” operation in VoteNet [6]; hence we name it the vote module. We further find that owing to the incomplete nature of the point cloud, one or more key points of the human body may not have been observed by the point cloud, leading to incomplete estimation of the template pose. To address this issue, we introduce a self-attention-based refinement module for completion and refinement, which consists of two self-attention layers. Notably, the refinement module does not have any point clouds or corresponding feature inputs. Instead, it relies on learned pose priors from the data.

Loss: We use the L2 loss to constrain the estimated pose J and ground truth human pose J^{gt} :

$$L_J = \|J - J^{gt}\|_2. \quad (9)$$

To directly constrain the per-point features, we constrain the output of the vote module. Specifically, for each point p_i in the input point cloud, we can obtain the ground truth values $\check{\mu}_i$ and $\check{q}_i$ from the ground truth pose. Then we use L2 loss and cross-entropy loss to constrain them:

$$L_\mu = \sum_{i=1}^N \|\mu_i - \check{\mu}_i\|_2, L_q = \sum_{i=1}^N \text{CrossEntropy}(Q_i, \check{Q}_i), \quad (10)$$

where μ_i and Q_i are the estimated values, $Q_i = (q_{i1}, \dots, q_{iJ})$, and $\check{Q}_i = (\check{q}_{i1}, \dots, \check{q}_{iJ})$, J is the number of key points. The overall loss function for the proposed PRN is as follows:

$$L_{PRN} = L_J + L_\mu + L_q, \quad (11)$$

B. Mesh Reconstruction Network

As illustrated in Fig. 5, we use heavy edge matching to down-sample the complete human body mesh a total of 9 times to obtain a set of human skeletal structures at different resolutions.

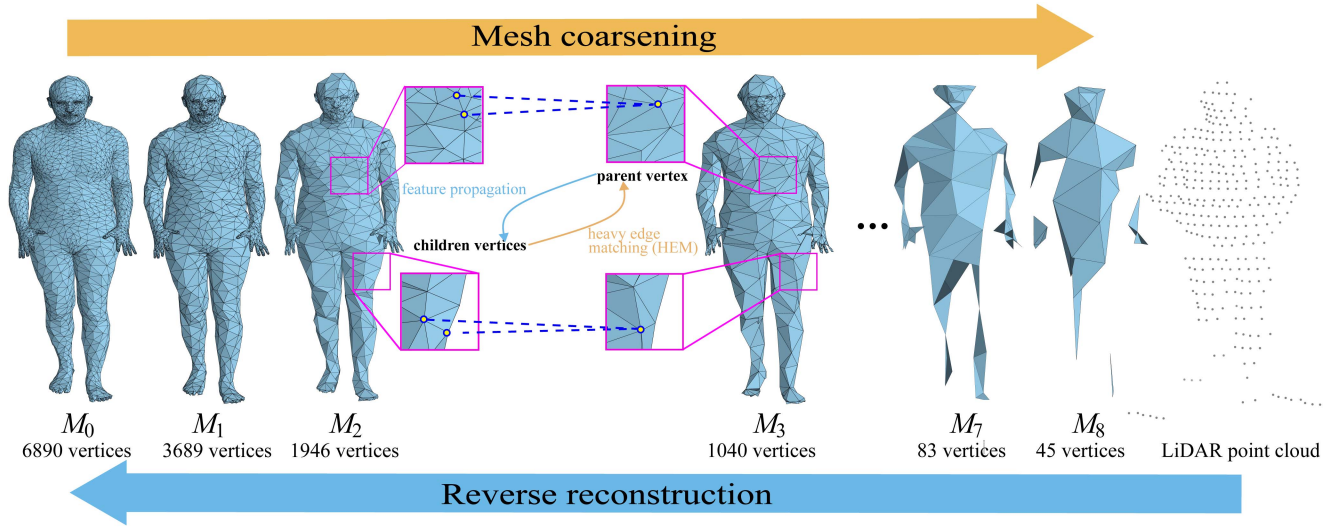


Fig. 5. The mesh coarsening gradually generates multiple coarse graphs with heavy edge matching (HEM) following [3]. Specifically, we perform reverse reconstruction with the MRN, which utilizes the parent-children relationship during coarsening. The vertex features are propagated following the parent-children edge to generate a higher-resolution mesh.

The sparsest skeleton corresponds to 24 vertices, whereas the densest mesh possesses 6890 vertices. This process constitutes the transition from sparse to dense. The MRN estimates the sparse vertices using the input template pose and point cloud features from PRN and gradually reconstructs the complete human body mesh. The overall structure of the MRN is illustrated in Fig. 4(b).

Graphormer for single resolution processing: The template pose is input into a fully connected layer that outputs sparse vertex features. We then utilize the graphormer propagation module which consists of a point cloud-based graphormer [27] and a propagation module. The point cloud-based graphormer consists of a cascade of self-attention layers followed by a graph convolutional layer. It is utilized to process the relationships between the point cloud and the vertices of a mesh. The vertex and point cloud features are concatenated and input into a self-attention layer to introduce fine local point cloud features. The features of the mesh vertices after the self-attention module are then input into a graph convolutional layer to model the link relationships between vertices. Notably, the initial point cloud features are derived from the PRN, and the point cloud features are updated after the self-attention module.

Propagation module for interresolution conduction: To obtain higher-resolution vertex features, Pose2Mesh [3] uses an upsampling approach, which, in this random way, does not consider the correspondence between vertices from different resolution meshes. We found that the correspondence and connecting relationships remain unchanged during the point cloud upsampling process. Specifically, as illustrated in Fig. 5, each parent vertex has 1-2 child vertices at a finer resolution. We model the point features between different resolution meshes on the basis of this correspondence, assuming that the feature of a child vertex is obtained by a specific offset from its parent vertex feature. This is consistent with the inverse process of downsampling. We use two different fully connected layers to model these offsets

between the relationships of the different resolution meshes, which is the propagation module.

For the first seven resolutions, we use one graphormer module and one propagation module for upsampling. For the last two resolutions with 3679 and 6890 vertices, owing to the large number of vertices, the use of a graphormer module results in an excessive computational burden. Therefore, we use two layers of fully connected layers to obtain the fine human mesh. Notably, we use a unified decoder to decode the vertex features, enabling each resolution of these mesh features to be decoded into the corresponding 3D coordinates. This ensures the consistency of the features learned by the MRN at different resolutions. Furthermore, we can supervise not only the final mesh output but also every intermediate resolution.

Loss: The final output mesh loss L_F is consistent with that in [3], which consists of vertex coordinate loss, joint coordinate loss, surface normal loss, and surface edge loss. In addition, we also apply a mesh loss based on intermediate resolution. In intermediate resolution i (from 1 to 8), L1 loss is utilized:

$$L_{v_i} = ||M_i - \tilde{M}_i||_1, \quad (12)$$

where M_i is the estimated mesh and $\tilde{M}_i$ is the corresponding ground truth mesh. Then we can obtain the overall mesh loss for intermediate resolutions:

$$L_{\text{inter}} = \sum_{i=1}^N L_{v_i}. \quad (13)$$

Finally, the entire network is trained end-to-end, so losses in the PRN are also involved in training:

$$L_{\text{MRN}} = L_{\text{PRN}} + L_F + L_{\text{inter}}. \quad (14)$$

C. MeshIK: Inverse Kinematics From 3D Human Mesh

Nonparametric algorithms such as MRN, Pose2Mesh [3] and MeshGraphormer [27] output mesh vertex positions without

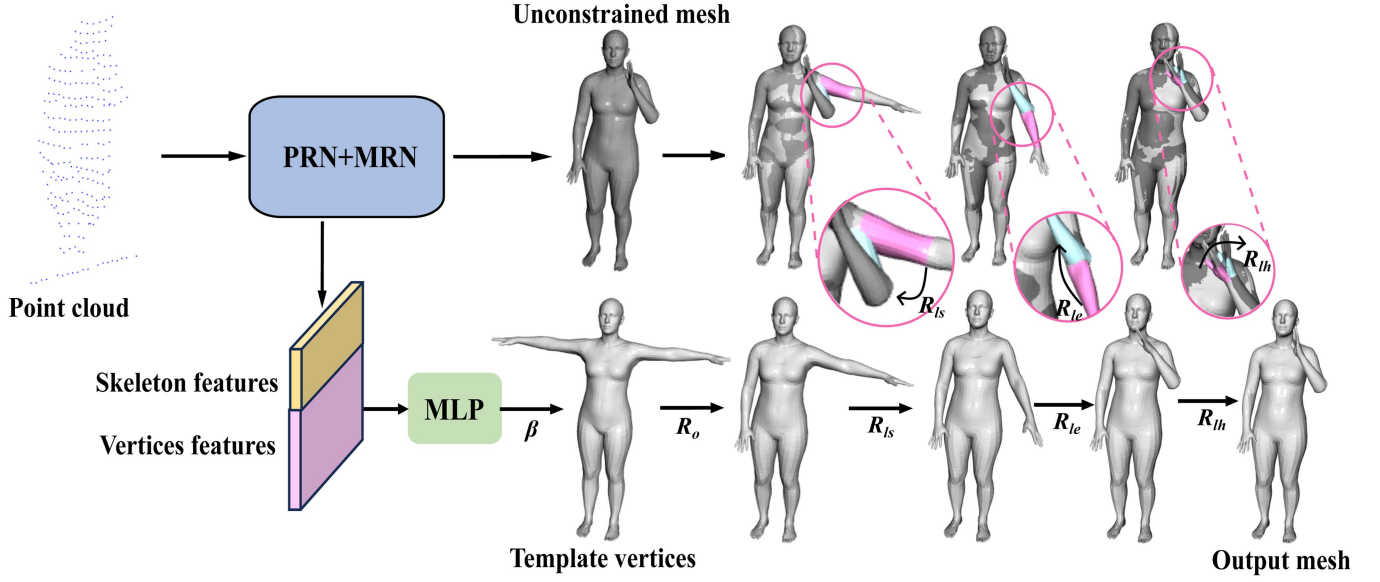


Fig. 6. MeshIK reconstructs the SMPL pose and shape parameters from the unconstrained mesh estimated by MRN (gray mesh). Specifically, the template vertices (light mesh) are overlapped with the unconstrained vertices for better visualization. MeshIK follows the inference order of the SMPL model and gradually regresses each rotation matrix from the root. Here, we present an example of a left arm with key points: the left shoulder, left elbow, and left hand. Pink vertices indicate the selected local vertices for SVD in the template pose and the corresponding vertices in the unconstrained mesh are in light blue. (R_{ls} : rotation matrix of the left shoulder; R_{le} : rotation matrix of the left elbow; R_{lh} : rotation matrix of the left hand; R_o : rotation matrices of all other poses, which are obtained in the same way as the aforementioned three key points.)

constraints, which could cause unsatisfactory local details. Here, we propose a local point matching-based algorithm, MeshIK, to estimate the SMPL [30] pose and shape parameters from the estimated mesh vertices. As illustrated in Fig. 6, we first utilize an MLP to estimate the SMPL shape parameters of the human body. With the shape parameters, we can obtain the SMPL template vertices. Leveraging the cascading skeleton-driven orders of the SMPL model, we can resolve the related rotation matrix to ensure that the local surface point coordinates of the variables and the local surface point coordinates of the templates match as closely as possible. Fortunately, this can be accomplished using singular value decomposition (SVD). Specifically, for the predicted local mesh vertices $v = [v_0, \dots, v_{N-1}]$ and local template vertices $t = [t_0, \dots, t_{N-1}]$, the local rotation matrix R_θ to be solved is given by:

$$R_\theta = \arg \min_R \sum_{i=0}^{N-1} w_i \|v_i - Rt_i\|_2^2, \quad (15)$$

where w_i is the vertexwise weight calculated from the SMPL model parameters. Note that:

$$\begin{aligned} \min \sum_{i=0}^{N-1} w_i \|v_i - Rt_i\|_2^2 &= \min \sum_{i=0}^{N-1} \|w_i p_i - R(w_i t_i)\|_2^2 \\ &\Leftrightarrow \min \|V_o - RT_o\|_F^2 \\ &\Leftrightarrow \min \text{trace}((V_o - RT_o)^T (V_o - RT_o)) \\ &\Leftrightarrow \max \text{trace}(V_o^T RT_o) \\ &\Leftrightarrow \max \text{trace}(RT_o V_o^T), \end{aligned} \quad (16)$$

where

$$\begin{aligned} V_o &= [w_0 v_0, w_1 v_1, \dots, w_{N-1} v_{N-1}], \\ T_o &= [w_0 t_0, w_1 t_1, \dots, w_{N-1} t_{N-1}]. \end{aligned} \quad (17)$$

$\|\cdot\|_F$ denotes the Frobenius norm, and V_o and T_o are weighted coordinate matrices. We can obtain the singular value decomposition of $T_o V_o^T$ as V and U :

$$T_o V_o^T = U D V^T, \quad (18)$$

where D is a diagonal matrix and each element of D , $d_{ii} \geq 0$. Then:

$$\begin{aligned} \max \text{trace}(RT_o V_o^T) &\Leftrightarrow \max \text{trace}(R U D V^T) \\ &\Leftrightarrow \max \text{trace}(D V^T R U). \end{aligned} \quad (19)$$

Note that $M = V^T R U$ is an orthogonal matrix. Hence each element of M , $|m_{ij}| \leq 1$. Then:

$$\text{trace}(D V^T R U) = \text{trace}(D M) = \sum_{i=1}^3 d_{ii} m_{ii} \leq d_{ii}. \quad (20)$$

The trace is maximized when $m_{ii} = 1, \forall 1 \leq i \leq 3$, where M is the identity matrix. Hence:

$$V^T R U = I \Rightarrow R_\theta = V U^T. \quad (21)$$

Algorithms and pseudo code: It is worth noting that owing to the rotation relationship of the human skeleton, the rotation matrix of the parent node will affect all its child nodes, hence the implementation varies with different poses. Specifically, the SMPL [30] model inference process follows the human skeleton tree of SMPL joints, and the global rotation matrices A_i utilized for the linear blend shape (LBS) of key point i are obtained

Algorithm 1: MeshIK in a PyTorch-Like Style**Input:** $M, T, W, Ind, R, R_r, Pa, A$ **Output:** R, θ $R[0] = R_r, A[0] = R_r$ **for** $i \leftarrow 1$ **to** $\text{len}(Pa) - 1$ **do** $M_- = M[Ind[i]], T_- = T[Ind[i]]$ $W_- = W[:, Ind[i]]$ $WA_P = (W_- * A) @ T_-$

Remove the impact of other key points.

 $P_0 = M_- - A[Pa[i]] @ WA_P$ $T_0 = W_-[i] * T_-$ $S = T_0 @ P_0$ $U, \Sigma, V = \text{SVD}(S)$ $A[i] = V @ U, R[i] = A[Pa[i]]^{-1} @ A[i]$ **end** $\theta = \text{Rotmat_to_axis_angle}(R)$

by multiplying all parent key points' local rotation matrices R . MeshIK determines R_i incrementally according to the skeleton tree. A simplified version of the pseudocode is illustrated in Algorithm 1, in which we omit the steps of matrix shape adjustment.

Specifically, M is the estimated mesh vertices with shape (6890, 3); T is the template mesh vertices estimated from the shape parameters, T possesses the same shape as M ; W is the LBS weights of the SMPL model with shape (24, 6890); Ind is the list of selected vertices index with bool type and length 24; R is the estimated local rotation matrix with shape (24, 3, 3), which is initialized with zero; R_r is the root rotation matrix estimated by the same method from HybrIK [25]; Pa is the parent index with shape (24), and for example, $Pa[3] = 0$; A is the LBS rotation matrix with shape (24, 3, 3), which is initialized with zero. θ is the estimated SMPL pose parameters with shape (72). $@$ indicates matrix multiplication and $*$ indicates elementwise multiplication.

Specifically, during implementation, M and T are normalized to the corresponding local coordinates with the current key points i , and finally, the utilized matrices $\hat{A}$ for the LBS with shape (24, 4, 4) are as follows:

$$\hat{A}_i = \begin{pmatrix} A_i & t_i \\ 0 & 1 \end{pmatrix}, \quad (22)$$

where t_i is the local shift of the template key points which can be calculated by A_i and the template skeleton length l_t .

Finally, the local vertices indices Ind are obtained for each joint by thresholding the LBS weights W . For each joint i and vertex j , we filter vertices that satisfy $W_{ij} > 0.85$ as selected vertices, which produces Ind .

Loss: With the input mesh of MeshIK M_{inp} , output M_{out} and the corresponding ground truth $\tilde{M}$, the overall loss of MeshIK is defined as follows:

$$L_{\text{MIK}} = L_{\text{MRN}} + ||M_{\text{inp}} - M_{\text{out}}||_2^2 + L_F(M_{\text{out}}, \tilde{M}). \quad (23)$$

The final output mesh loss L_F is consistent with that in [3].

Essentially, MeshIK establishes the relationship between human mesh vertices and SMPL pose parameters, and equates the task of learning the SMPL pose parameters with the regression mesh vertices, which greatly reduces the learning difficulty in the point cloud scenario.

D. Implementation Details

The number of input points is set to 1024. The entire LiDAR-HMR is end-to-end trainable. In the actual training process, we divide the training process into two steps to achieve better convergence. First, we pretrain the PRN with a batch size of 64 for 50 epochs with L_{PRN} until convergence. Then, we perform end-to-end training of the PRN, MRN, and MeshIK together in a batch size of 8 for 50 epochs with L_{MIK} . The network parameters are updated via Adam [19], and the learning rate is set to 5×10^{-4} . Specifically, the learning rate is reduced by a factor of 2 at the 10th and 20th epochs for the subsequent end-to-end training.

IV. EXPERIMENTS**A. Datasets**

We conduct our experiments on four public datasets: Waymo [36], Human-M3 [7], SLOPER4D [5] and LiDARHuman26M [24].

The Waymo open dataset [36] releases the human key point annotations on the v1.3.2 dataset, which contains LiDAR range images and associated camera images. We use v2.0 for training and validation. It possesses 14 key points annotation for each object. There are 8125 annotated human key points in the training set and 1873 in the test set.

The Human-M3 dataset [7] is a multimodal dataset that captures outdoor multiperson activities in four different scenes. It possesses 15 key point annotations for each object. There are 80103 annotated human key points in the training set and 8951 in the test set.

The SLOPER4D dataset [5] is an outdoor human motion capture dataset that captures several objects on the basis of LiDAR point clouds and RGB images. Ground-truth meshes are obtained via motion capture devices. Overall, six motion fragments exist in the released part. As there is no manual assignment of training and test sets by the author, we selected a fragment as the test set. As a result, there are 24936 annotated human meshes in the training set and 8064 in the test set.

The LiDARHuman26M dataset [24] is a human motion capture dataset that captures several objects on the basis of LiDAR point clouds and RGB images. Ground-truth meshes are obtained via motion capture devices. There are approximately 160 K annotated human meshes in the training set and 24 K in the test set.

Owing to the lack of 3D mesh annotation in the Waymo and Human-M3 datasets, we used key point annotations and input point clouds to reconstruct the pseudolabel of the human mesh. This process is similar to that of Smplify-X [34], named Smplify-Cloud. For input point cloud Q and ground-truth joint annotation $P = [p_1, \dots, p_N]$, N is the number of points.

TABLE I
EVALUATION OF SEMPLIFY-CLOUD IN THE SLOPER4D DATASET

SLOPER4D	MPJPE	MPVPE	MPERE
Smplify-Cloud	1.06	2.26	0.061

The metrics include MPJPE (cm), MPVPE (cm), and MPERE.

We estimate the pose parameter θ and shape parameter β of the SMPL [30] model and the corresponding root translation r . Specifically, pseudolabels are achieved by minimizing the loss function that constrains observations and priors:

$$[r_p, \theta_p, \beta_p] = \arg \min_{r, \beta, \theta} (L_{op}), \quad (24)$$

$$P_M, M = S_-(r, \beta, \theta), \quad (25)$$

$$L_{op} = \lambda_J L_J(P_M, P) + \lambda_p L_p(M, \beta, \theta), \quad (26)$$

$$L_p(M, \beta, \theta) = L_{sp}(\beta) + L_{pp}(\theta) + L_{cp}(M), \quad (27)$$

where S_- indicates SMPL model inference, M indicates the 3D mesh vertices and P_M indicates the 3D human pose joints processed by the SMPL model. L_J is the l2-loss between the estimated 3D joints and the annotation joints. L_p is the prior loss inherited from Smplify-X [34]. Specifically, L_{sp} constrains the human shape, L_{pp} constrains the human pose and angle, L_{cp} is the collision punishment for the estimated mesh, and λ_J and λ_p are hyperparameters. The pseudolabels can be obtained via the L-BFGS optimization algorithm.

Specifically, to check the quality of the generated pseudolabels, we use Smplify-Cloud to generate pseudolabels on the SLOPER4D dataset and evaluate them using ground-truth annotation. The evaluation results are illustrated in Table I, with metrics at a low level, so the quality of the pseudolabels is guaranteed.

For the Waymo-v2 and Human-M3 datasets, we used the annotated root joints (pelvis joints) of each pedestrian as the center and selected the point clouds contained within a cube of 2 m side length as the input to the network. When applied to large scene data, we expect that the 3D positions of pedestrians can be estimated by existing state-of-the-art pedestrian localization networks. Specifically, to simulate the estimation errors involved in pedestrian localization in practical applications, as well as random rotations, we apply random translations in the range of $[-0.2 \text{ m}, 0.2 \text{ m}]$ and random rotations in the range of $[-0.25\pi, 0.25\pi]$ during the training of the network.

B. Evaluation Metrics

For HPE, we utilize MPJPE [14] and PA-MPJPE [3], [10], and the mean per vertex position error (MPVPE) for HMR. In addition, we propose a new metric, the mean per edge relative error (MPERE), for nonparametric method evaluation. For nonparametric methods, the position relationships between vertices are not fixed, and MPERE can represent the reconstruction quality of the local details.

For a predicted mesh M with edge length set $(l_1, \dots, l_m)$, and ground-truth mesh $\tilde{M}$ with edge length set $(\tilde{l}_1, \dots, \tilde{l}_m)$, the

MPERE is defined as follows:

$$MPERE = \sum_{i=1}^m \frac{1}{m} \frac{\|l_i - \tilde{l}_i\|_1}{\tilde{l}_i}, \quad (28)$$

where m is the number of edges in the mesh. MPERE measures the ratio of the length error to the ground-truth length of mesh edges and judges the reconstruction quality of short edges in dense parts more efficiently. As illustrated in Fig. 7, MPVPE may not be enough to measure the reconstruction quality. In this case, MPERE is needed as an additional measure.

In addition, we evaluated the computational cost during network training (GFLOPs) and the throughput during inference (frames per second, fps) of all the algorithms to compare the efficiency and lightness of the algorithms. Specifically, we use a GeForce RTX 3090 GPU as platform and record the network inference speed with a batch size of 1 in the test set of the Waymo-v2 dataset. For V2V-PoseNet and LPFormer, we follow the original setting where the volume cube size is set to $88*88*88$.

C. Comparison to the State-of-the-Art

For 3D HPE, we compare V2V-Posenet [31] (V2V) and reimplement the LPFormer [44] (LP). For 3D HMR, we compare Pose2Mesh [3] (P2M), HybriK [25], LiDARCap [24] (with single-frame input), reimplemented SAHSR [16], and VoteHMR [28].

To evaluate the role of the PRN as a point cloud extraction backbone, we design and utilize the settings of LP+MRN+MeshIK for comparison. The LPFormer is pre-trained, followed by end-to-end training of the entire network, as consistent with LiDAR-HMR. In addition, Pose2Mesh requires a well-estimated 3D human skeleton; hence, we utilize the results from the best 3D HPE model in the corresponding datasets as the input pose, called “P2M”. For a fair comparison, we also utilize our proposed PRN (pretrained) to concatenate with MeshNet in the Pose2Mesh network for end-to-end training (the same as LiDAR-HMR), called “PRN+P2M” and “PRN+P2M+MeshIK”.

The quantitative evaluation results are illustrated in Table II. In particular, for **HPE**, the PRN achieves comparable results to those of state-of-the-art methods on different datasets but with significantly lower computational requirements. They rely on the voxelization of point clouds and 3D CNNs for extracting 3D features, which results in some computational redundancy and consumes more computational resources. Furthermore, the 3D features extracted by 3D CNNs do not provide significant assistance in the subsequent human mesh recovery. In addition, the reconstruction performance of “V2V+P2M” is weaker than that of the comparative group “PRN+P2M”. This also demonstrates the effectiveness of our PRN network. The performance comparison between “LP+MRN+MeshIK” and “LiDAR-HMR” shows that although LPFormer can achieve good performance for pose estimation alone, when it is used as a part of the mesh reconstruction network, only the output of pose skeleton is not sufficient. Accordingly, the PRN can effectively extract effective features

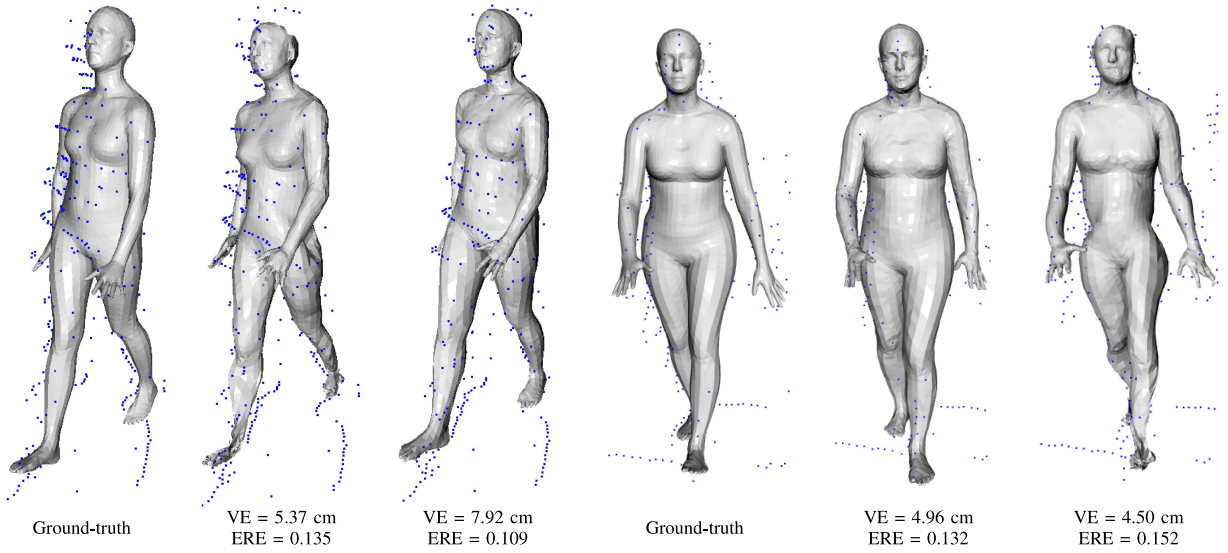


Fig. 7. Two examples of estimated human meshes, and point clouds are illustrated by the blue points. Meshes with low VE (MPVPE) may not have the best reconstruction quality. Therefore, we introduce the ERE (MPERE) to measure the reconstruction results from another perspective.

TABLE II
EVALUATION RESULTS ON THE SLOPER4D, WAYMO, HUMAN-M3 AND LiDARHUMAN26 M DATASETS

	Model	SLOPER4D			Waymo			Human-M3			LiDARHuman26M			GFLOPs	FPS
		JE	VE	ERE	JE	VE	ERE	JE	VE	ERE	JE	VE	ERE		
Pose	V2V	5.07	-	-	7.03	-	-	8.30	-	-	8.78	-	-	61.803	22.51
	LP	7.71	-	-	6.39	-	-	7.75	-	-	7.93	-	-	62.197	23.92
	PRN	5.70	-	-	6.78	-	-	8.22	-	-	8.46	-	-	0.672	23.92
Mesh	HybrIK	5.63	8.82	0.097	6.96	9.00	0.073	8.53	12.06	0.096	10.66	15.97	0.125	0.780	12.56
	SAHSR	7.26	8.12	0.085	9.66	11.68	0.163	10.55	13.15	0.291	10.70	13.34	0.111	0.427	24.39
	VoteHMR	5.46	6.09	0.079	9.76	12.08	0.194	10.58	12.54	0.189	13.14	16.22	0.211	0.663	21.81
	LiDARCap	9.79	10.56	0.06	11.10	13.19	0.061	10.12	11.51	0.032	9.35	12.52	0.036	0.522	77.32
	V2V+P2M	5.07	5.98	0.126	9.77	8.56	0.150	10.44	9.35	0.111	12.88	13.13	0.113	3.756	13.61
	PRN+P2M	5.66	6.53	0.132	8.74	9.04	0.150	8.06	8.96	0.091	7.44	10.20	0.113	4.096	14.38
	PRN+P2M+MeshIK	6.36	6.27	0.042	7.65	9.24	0.067	8.12	9.02	0.034	7.54	10.18	0.089	4.180	7.55
	LP+MRN+MeshIK	4.31	5.46	0.063	6.86	8.82	0.067	7.84	9.22	0.047	7.57	10.39	0.045	63.854	7.52
	PRN+MRN	4.76	5.19	0.094	7.01	8.24	0.119	7.94	8.95	0.088	7.63	10.39	0.079	2.225	12.35
	LiDAR-HMR	4.77	4.97	0.039	6.62	7.80	0.062	7.60	8.64	0.044	7.62	10.25	0.075	2.309	7.45

The metrics include JE (MPJPE, cm), VE (MPVPE, cm), and ERE (MPERE), together with the computation costs (GFLOPs) while training and the inference efficiency (FPS) when batch size is set to 1. The proposed methods are shown in bold and the best values are shown in bold.

from the input point cloud, which is helpful for the subsequent mesh reconstruction.

For **HMR**, the qualitative comparison and quantitative evaluation results are illustrated in Fig. 8 and Table II, respectively. Both HybrIK and LiDARCap reconstruct the SMPL pose parameters on the basis of the skeleton, but they suffer from inaccurate local details, which are generally reflected by the unfaithful axial rotations of the bones under the correct skeleton. This is mainly because they fail to extract precise local details from the point clouds. VoteHMR and SAHSR do not achieve satisfactory results. Both methods directly regress the human body mesh from the point clouds without sparse-to-dense modeling. They are parameterized methods that face difficulties in effectively modeling the relationships between point clouds and corresponding human meshes. For the more challenging datasets such as Waymo and Human-M3, larger errors result in unsatisfactory MPERE scores. Comparing the above two methods highlights the effectiveness of the proposed point

cloud-to-SMPL reconstruction pipeline. Specifically, LiDAR-HMR also performs well in reconstructing more complex and diverse poses, demonstrating adequate robustness and portability. The final throughput of the network is approximately 7.5 frames per second (fps), which is close to that of Pose2Mesh+MeshIK. In particular, when the MeshIK network structure is excluded, the network's throughput capability is 12.35 fps, which is slightly lower than that of Pose2Mesh's 14.38 fps. This can be attributed mainly to the deeper and more complex feature processing introduced by the MRN. However, it is important to note that the typical data frame rate of commonly used LiDAR systems is approximately 10 fps. Therefore, the network's throughput capability can meet the application requirements.

For Pose2mesh, although both methods are based on a similar pipeline of sparse-to-dense modeling, the MRN takes the point cloud as input to assist in the reconstruction. With the same backbone and training conditions, the MRN outperforms the MeshNet proposed in Pose2Mesh while consuming fewer

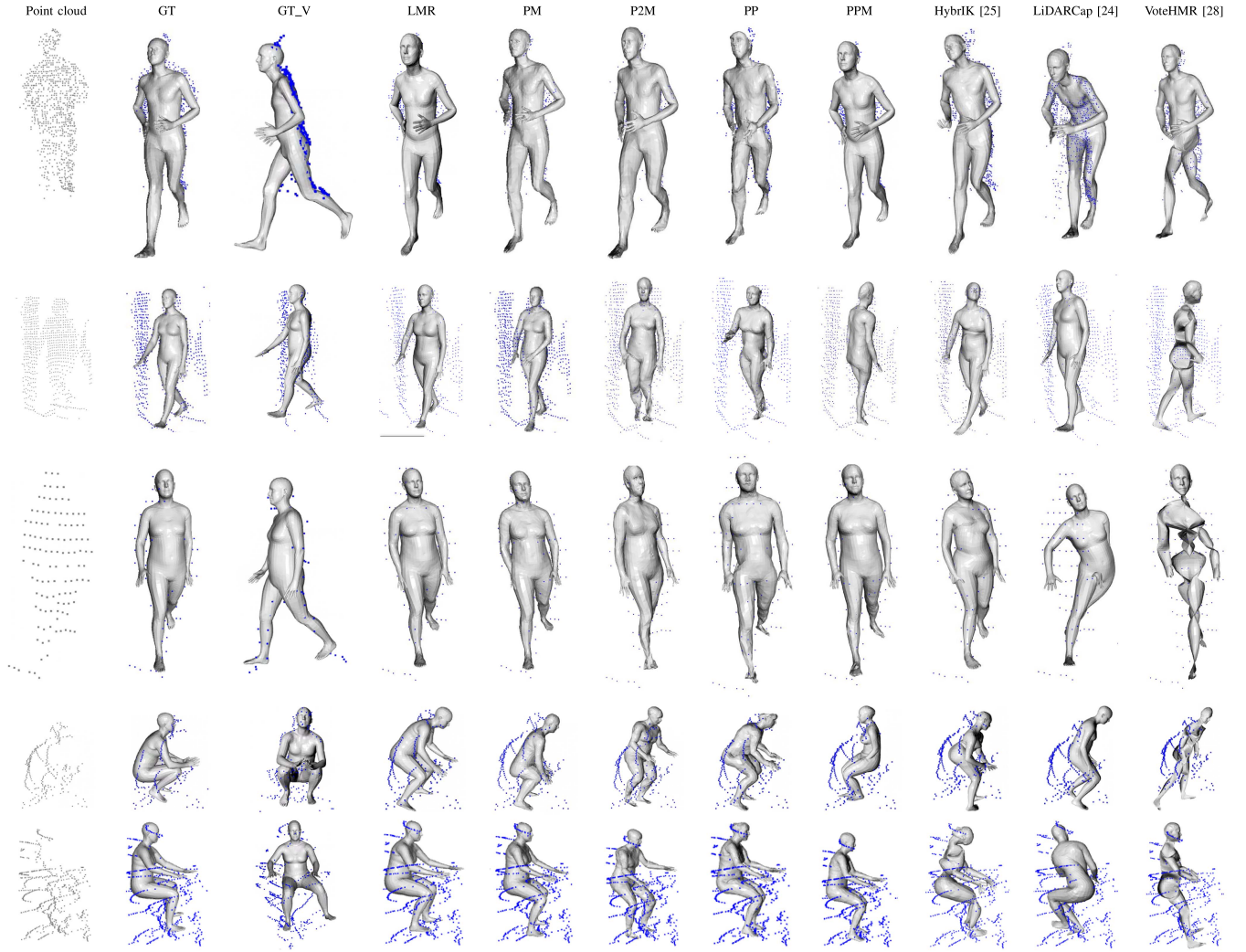


Fig. 8. Comparisons with the state-of-the-art methods on the Waymo, SLOPER4D and Human-M3 datasets. The meshes are visualized with the input point clouds (blue points) for better comparison. The top three rows show examples corresponding to challenges of incomplete point clouds (low leg missing), noise, and sparse point clouds. The bottom two rows show more diverse human poses that exist in the Human-M3 dataset. (GT: ground-truth, GT_V: ground-truth (alternative view), LMR: LiDAR-HMR (PRN+MRN+MeshIK), PM: PRN+MRN, P2M: Pose2Mesh [3], PP: PRN+P2M, and PPM: PRN+P2M+MeshIK).

computational resources. Additionally, constraints are applied to the intermediate reconstruction results, making the entire reconstruction process more rational and effective. As illustrated in Fig. 8, what cannot be quantitatively measured from the metrics is that Pose2Mesh often suffers from unreasonable body local details. Although the Pose2Mesh performances on the LiDARHuman26 M dataset are better in MPJPE and MPVPE than that of LiDAR-HMR, their higher MPERE also indicates a common irrational situation of regressed human poses. This is mainly because the connection relationship of vertex points is not considered in the upsampling process, which makes it difficult for the network to model the semantic information of each vertex point.

Finally, the meaning of **MeshIK** is clear, and the rationality and accuracy of the estimated poses are improved after MeshIK's processing. This is reflected in MPERE's improvement and more reasonable visualization of the results. After MeshIK, the original vertex coordinate output of the model is transformed into semantically SMPL shape and pose parameters, which describe

the posture of the human body. Compared with the original representation, this representation can be used for other tasks such as timing processing or motion recognition. In general, the proposed LiDAR-HMR achieves the best results with fewer computing resources under four datasets and different settings.

D. Ablation Study

We conduct ablation studies on the Waymo dataset [36].

Ablation of the PRN: We compare the following different settings: (1) Without the vote module, the output features of PointTransformer-v2 are fed into a fully connected layer to directly output the human pose. (called “no_vote”) (2) The refinement module is removed. (called “no_refine”) (3) With the condition of “no_vote”, the refinement module is removed. (called “no_all”) (4) The loss terms of L_μ and L_q are removed from L_{PRN} in (11) (called “no_voteloss”).

The quantitative evaluation results are listed in Table III. Compared with the “no_all” setting, the introduction of the vote

TABLE III
ABLATION EVALUATION OF THE PRN ON THE WAYMO DATASET

Group	MPJPE	PA-MPJPE	GFLOPs
no_vote	6.89	5.07	0.751
no_refine	7.05	5.62	0.475
no_all	10.14	7.31	0.347
no_voteless	6.94	5.09	0.672
PRN	6.78	5.20	0.672

The metrics include MPJPE (cm), PA-MPJPE (cm), and the computation costs while training when the batch size is set to 1 (GFLOPs). The best values are shown in bold.

module results in an MPJPE improvement of 3.09 cm and a PA-MPJPE improvement of 1.69 cm. Similarly, adding the refinement module can also result in improvements of 3.25 cm and 2.24 cm respectively. Notably, both the refinement module and the vote module can achieve good results when used alone. They obtain better human poses from the two aspects of “more accurate estimation” and “more accurate adjustment”. Even so, the introduction of the vote module can still reduce the computational consumption of the network. The vote loss term in (11) has a slight benefit on the performance of the PRN, and it is more natural and direct for the supervised backbone to learn point-by-point features.

PA-MPJPE measures only the relative error of human skeletons, reflecting the network’s ability to perceive overall human poses. Intuitively, under the “no_vote” setting, the network lacks access to more local human detail, thus focusing more on reconstructing the relative 3D skeleton as a whole; this leads to improved performance in PA-MPJPE. However, it is important to note that PA-MPJPE does not account for errors in the location of the body in global coordinates. In practical applications, especially in the subsequent human mesh reconstruction process, obtaining accurate 3D skeletons in global coordinates is highly important.

Ablation of MRN: We compare the performance of the MRN in three different aspects: the utilization of raw point cloud features from the PRN, different structures of the pose-to-mesh (P2M) reconstruction process, and different loss functions of the MRN. Specifically, for the utilization of the PRN features we utilize the following settings: (1) The PRN is initialized with pre-trained weights and no longer participates in end-to-end training, and the loss function does not include the pose estimation loss of the PRN (called “no_PRN”). (2) The PRN is not pretrained and initialized with random weights (called “no_pretrain”). (3) The point cloud features are not fed into the reconstruction process of the MRN, and the self-attention modules at each resolution only calculate the correspondence between vertex features (called “no_pcd”). For P2M structures, we utilize the following settings: (1) The output features of the PRN are directly fed into an MLP-based network to regress the SMPL pose and shape parameters (called “no_MRN”). (2) The output features of the PRN are fed into a graphormer-based network consistent with that in [27] to upsample and regress vertex coordinates (called “graphormer”). (3) Different resolutions in the MRN network use upsampling (consistent with Pose2Mesh) instead

TABLE IV
ABLATION EVALUATION OF THE MRN ON THE WAYMO DATASET

	Group	MPJPE (PRN)	MPVPE	MPERE	GFLOPs
PRN Features	no_PRN	-	28.03	0.306	2.225
	no_pretrain	6.62	8.53	0.128	2.225
	no_pcd	7.05	9.05	0.145	2.103
P2M Structures	no_MRN	6.86	10.22	0.057	0.750
	graphormer	8.05	9.83	1.760	9.150
	upsample	6.49	8.34	0.127	2.164
MRN Losses	no_mid	6.53	8.89	0.123	2.212
	edge_mid	6.31	8.31	0.165	2.225
PRN+MRN		6.28	8.24	0.119	2.225

The metrics include MPJPE (cm), MPVPE (cm), MPERE, and the computation cost while training when the batch size is set to 1 (GFLOPs). The best values are shown in bold.

of feature propagation (called “upsample”). For losses, we utilized the following settings: (1) The loss function does not include the mesh reconstruction loss at intermediate resolutions (called “no_mid”), and (2) the reconstruction loss at intermediate resolutions not only constrains the corner point positions but also introduces constraints on edge lengths (L_F term in Section III-B) (called “edge_mid”).

The quantitative evaluation results are illustrated in Table IV. For the utilization of raw point cloud features from the PRN, first, end-to-end training is essential for the MRN. When the backbone does not participate in end-to-end training, the performance of the MRN decreases drastically, which indicates that the features learned from pose estimation pretraining cannot represent 3D mesh reconstruction. Therefore, weight updating is important for the backbone of MRN training. Second, the effect of using raw point cloud features is obvious; when comparing the settings “PRN+MRN” and “no_pcd”, the MPVPE and MPERE decreased by 0.81 cm and 0.026, respectively, accounting for 8.95% and 17.93%. This demonstrates the effectiveness of the proposed strategy of integrating point clouds into the reconstruction process. Finally, the pre-training process is also beneficial for mesh reconstruction; the MPVPE and MPERE decreased by 0.29 cm and 0.009, respectively, accounting for 3.39% and 7.03%. This finding shows that the pretraining strategy can provide more semantic and prior information for mesh reconstruction, which is helpful.

Regarding the **pose-to-mesh network structures**: First, sparse-to-dense modeling is essential. Regressing parameters directly from the PRN features results in unsatisfactory performance. Second, inside sparse-to-dense reconstruction, the proposed approach of feature propagation in different mesh resolutions showed improvements over the upsampling method in Pose2Mesh and Graphormer, with MPVPE and MPERE decreasing by 0.1 cm and 0.008, respectively, compared with those in Pose2Mesh. This shows that the proposed feature propagation module inherits the information between different resolutions more effectively in the process of feature conduction, and achieves better results.

For the **MRN losses** of intermediate meshes, it was observed that supervising the mesh at intermediate resolutions has a slight effect on the algorithm’s performance. This may be attributed

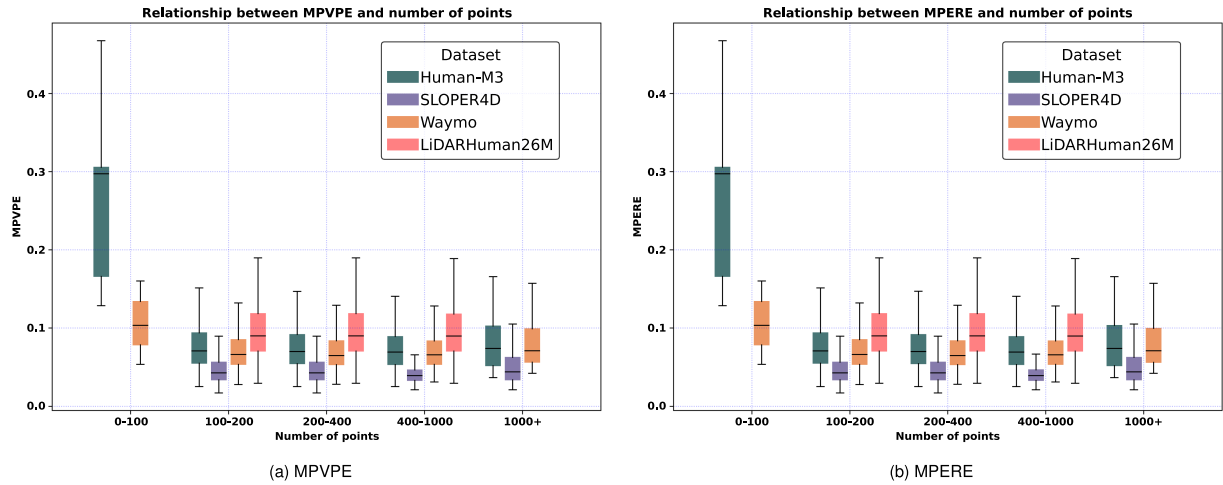


Fig. 9. Analysis of the relationship between performance and the number of points.

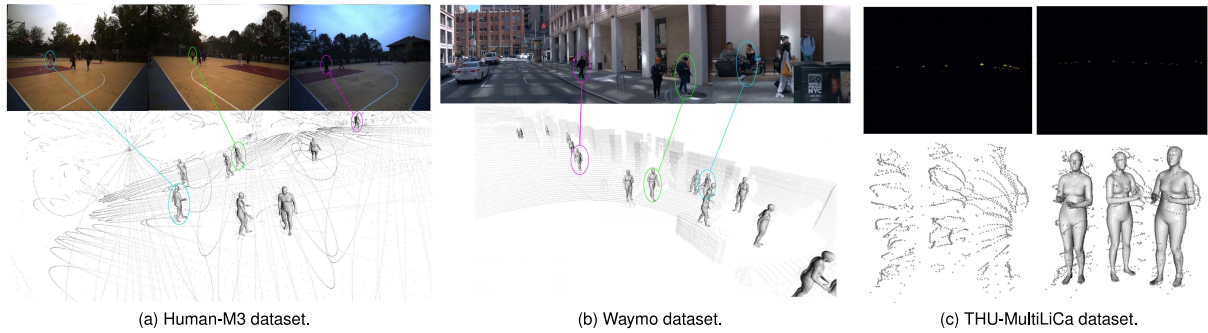


Fig. 10. More visual results of LiDAR-HMR in multiperson scenes. (a) Human-M3 dataset. (b) Waymo dataset. (c) THU-MultiLiCa [49] dataset (top: RGB images; bottom left: point clouds; bottom right: estimated meshes overlapping with the point clouds).

to providing the model with a more reasonable reconstruction process at intermediate stages, facilitating better learning. Interestingly, imposing constraints on the mesh length at intermediate resolutions has a mildly negative effect on the model performance, possibly because excessive constraints during the intermediate reconstruction process lead to reduced freedom in the final mesh reconstruction.

Analysis of the number of points: As shown in Fig. 9, the performance of LiDAR-HMR is relatively stable when the number of points exceeds 100. When the number of points is less than 100, the algorithm exhibits significant fluctuations in MPVPE and MPERE on the Human-M3 and Waymo datasets, respectively. The performance of LiDAR-HMR is indeed affected when the input points are sparse, and problems under such circumstances are more challenging. We found that the three public datasets have a limited number of samples with sparse points (less than 100), which may be a contributing factor to this phenomenon as the learning process is not sufficiently comprehensive in handling such cases. This issue can be addressed by extracting challenging samples.

E. Limitations and Failure Cases

Failure cases are illustrated in Fig. 11. Specifically, although the nonparametric representation can reconstruct the human

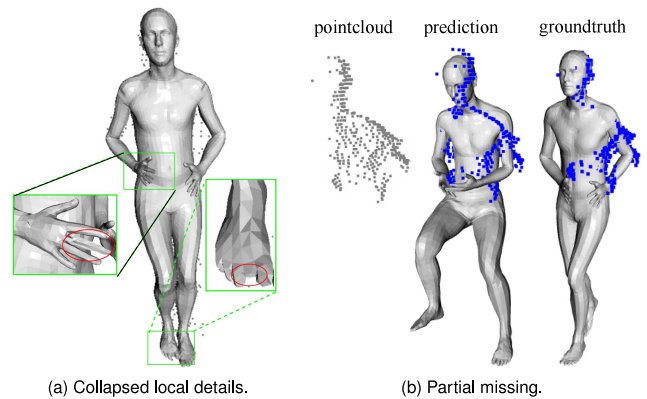


Fig. 11. Two kinds of typical failure cases. In (a), red ellipses mark the collapsed details.

mesh more accurately and intuitively than the parametric representation can, the mesh vertices regressed by the MRN still suffer from unsatisfactory local details which are collapsed and not smooth in some cases. This makes the reconstructed mesh unrealistic (Fig. 11(a)).

In addition, the incompleteness of the input point cloud can lead to errors in the overall mesh. LiDAR-HMR is capable of mitigating the issues arising from missing portions of the point

cloud to some extent, particularly through the network's ability to reconstruct the missing parts of the pose on the basis of effective pose priors learned from the training data. However, when a certain body part is entirely missing, LiDAR-HMR still struggles to achieve fully accurate results. Specifically, in Fig. 11(b), the lower half of the body is almost completely absent, which consequently leads to inaccurate estimations via LiDAR-HMR. Such challenging cases are indeed present in real-world applications and are difficult to avoid. Unfortunately, these cases are ambiguous, and introducing additional temporal constraints for pose continuity may be beneficial in addressing these types of issues.

Notably, valid human mesh priors (such as VPoser [34]) are based on parametric representations. Suppose that more human mesh priors can be introduced into network training, or that restrictions can be imposed on the surface mesh of the network output. This approach results in a more reasonable, realistic human surface while ensuring accuracy.

F. More Visual Results

Apart from Fig. 3, more visualization results of LiDAR-HMR in wide outdoor scenes are illustrated in Fig. 10. LiDAR-HMR offers robust and satisfactory results on different outdoor scenes, where HMR methods based on RGB images usually struggle. As illustrated in Fig. 10(c), the illumination is too weak to distinguish human bodies in the night scene of the THU-MultiLiCa [49] dataset, but LiDAR-HMR is not affected. Specifically, this dataset does not provide human pose ground-truth values, so we use the trained weights in the Waymo-v2 dataset for inference, which also shows the robustness and portability of LiDAR-HMR in different scenes.

V. CONCLUSION

We present the first attempt at 3D human pose estimation and mesh recovery on the basis of single-frame sparse LiDAR point clouds. We solve this problem by proposing a sparse-to-dense reconstruction pipeline. We introduce the input point cloud to assist the whole reconstruction process, which imposes constraints on the intermediate results of the reconstruction and achieves good results. In addition, we propose MeshIK to regress the SMPL parameters from unconstrained mesh vertices, which not only improves the rationality of the meshes but also imparts semantics to the output mesh. However, the proposed method still fails in some very sparse scenarios and can output unreasonable self-interspersed meshes. A possible solution is to consider more human poses a priori during the reconstruction process. We hope that our work will inspire subsequent work on sparse point clouds or multimodal human perception tasks.

REFERENCES

- [1] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2D pose estimation using part affinity fields," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7291–7299.
- [2] A. Chen et al., "Towards weather-robust 3D human body reconstruction: Millimeter-wave radar-based dataset, benchmark, and multi-modal fusion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 35, no. 1, pp. 273–286, Jan. 2025.
- [3] H. Choi, G. Moon, and K.M. Lee, "Pose2Mesh: Graph convolutional network for 3D human pose and mesh recovery from a 2D human pose," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 769–787.
- [4] P. Cong et al., "Weakly supervised 3D multi-person pose estimation for large-scale scenes based on monocular camera and single LiDAR," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 461–469.
- [5] Y. Dai et al., "SLOPER4D: A scene-aware dataset for global 4D human pose estimation in urban environments," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 682–692.
- [6] Z. Ding, X. Han, and M. Niethammer, "VoteNet: A deep learning label fusion method for multi-atlas segmentation," in *Proc. 22nd Int. Conf. Med. Image Comput. Comput.-Assist. Intervention*, Shenzhen, China, 2019, pp. 202–210.
- [7] B. Fan, S. Wang, W. Zheng, J. Feng, and J. Zhou, "Human-M3: A multi-view multi-modal dataset for 3D human pose estimation in outdoor scenes," 2023, *arXiv:2308.00628*.
- [8] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun, "Real-time human pose tracking from range data," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 738–751.
- [9] R. Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. Fitzgibbon, "Efficient regression of general-activity human poses from depth images," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 415–422.
- [10] J. C. Gower, "Generalized procrustes analysis," *Psychometrika*, vol. 40, pp. 33–51, 1975.
- [11] A. Haque et al., "Towards viewpoint invariant 3D human pose estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 160–177.
- [12] T. Helten et al., "Personalization and evaluation of a real-time depth-based full body tracker," in *Proc. 2013 Int. Conf. 3D Vis.*, 2013, pp. 279–286.
- [13] J. Hu, H. Zhang, Y. Wang, M. Ren, and Z. Sun, "Personalized graph generation for monocular 3D human pose and shape estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 4, pp. 2399–2413, Apr. 2024.
- [14] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments," *IEEE Trans. Pattern Anal. Mach. Intell.* vol. 36, no. 7, pp. 1325–1339, Jul. 2014.
- [15] H. Jang, M. Kim, J. Bae, and Y. M. Kim, "Dynamic mesh recovery from partial point cloud sequence," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 15074–15084.
- [16] H. Jiang, J. Cai, and J. Zheng, "Skeleton-aware 3D human shape reconstruction from point clouds," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 5431–5441.
- [17] A. Kamel, B. Sheng, P. Li, J. Kim, and D. D. Feng, "Hybrid refinement-correction heatmaps for human pose estimation," *IEEE Trans. Multimedia*, vol. 23, pp. 1330–1342, 2021.
- [18] A. Kanazawa, M. J. Black, D.W. Jacobs, and J. Malik, "End-to-end recovery of human shape and pose," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7122–7131.
- [19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
- [20] S. Knoop, S. Vacek, and R. Dillmann, "Sensor fusion for 3D human body tracking with an articulated 3D body model," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2006, pp. 1686–1691.
- [21] M. Kocabas, N. Athanasiou, and M. J. Black, "VIBE: Video inference for human body pose and shape estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5253–5263.
- [22] N. Kolotouros, G. Pavlakos, M. J. Black, and K. Daniilidis, "Learning to reconstruct 3D human pose and shape via model-fitting in the loop," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 2252–2261.
- [23] N. Kolotouros, G. Pavlakos, and K. Daniilidis, "Convolutional mesh regression for single-image human shape reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4501–4510.
- [24] J. Li et al., "LiDARCap: Long-range markerless 3D human motion capture with LiDAR point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 20502–20512.
- [25] J. Li et al., "HybriK: A hybrid analytical-neural inverse kinematics solution for 3D human pose and shape estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3383–3393.
- [26] K. Lin, L. Wang, and Z. Liu, "End-to-end human pose and mesh reconstruction with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 1954–1963.
- [27] K. Lin, L. Wang, and Z. Liu, "Mesh graphormer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 12939–12948.
- [28] G. Liu, Y. Rong, and L. Sheng, "VoteHMR: Occlusion-aware voting network for robust 3D human mesh recovery from partial point clouds," in *Proc. 29th ACM Int. Conf. Multimedia*, 2021, pp. 955–964.

- [29] H. Liu et al., "Fast human pose estimation in compressed videos," *IEEE Trans. Multimedia*, vol. 25, pp. 1390–1400, 2023.
- [30] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *Seminal Graph. Papers, Pushing Boundaries*, vol. 2, pp. 851–866, 2023.
- [31] G. Moon, J. Y. Chang, and K. M. Lee, "V2V-PoseNet: Voxel-to-voxel prediction network for accurate 3D hand and human pose estimation from a single depth map," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5079–5088.
- [32] G. Moon, J. Y. Chang, and K. M. Lee, "Camera distance-aware top-down approach for 3D multi-person pose estimation from a single RGB image," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 10133–10142.
- [33] M. Omran, C. Lassner, G. Pons-Moll, P. Gehler, and B. Schiele, "Neural body fitting: Unifying deep learning and model based human pose and shape estimation," in *Proc. Int. Conf. 3D Vis.*, 2018, pp. 484–494.
- [34] G. Pavlakos et al., "Expressive body capture: 3D hands, face, and body from a single image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10975–10985.
- [35] J. Shotton et al., "Real-time human pose recognition in parts from single depth images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 1297–1304.
- [36] P. Sun et al., "Scalability in perception for autonomous driving: Waymo open dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2446–2454.
- [37] H. Y. Tung, H. W. Tung, E. Yumer, and K. Fragkiadaki, "Self-supervised learning of motion capture," in *Proc. Adv. Neural Inf. Process. Syst.*, pp. 5242–5252, 2017.
- [38] K. Wang, J. Xie, G. Zhang, L. Liu, and J. Yang, "Sequential 3D human pose and shape estimation from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7275–7284.
- [39] Z. Weng et al., "3D human keypoints estimation from point clouds in the wild without human labels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 1158–1167.
- [40] X. Wu, Y. Lao, L. Jiang, X. Liu, and H. Zhao, "Point transformer V2: Grouped vector attention and partition-based pooling," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 33330–33342.
- [41] Y. Wu, D. Kong, S. Wang, J. Li, and B. Yin, "An unsupervised real-time framework of human pose tracking from range image sequences," *IEEE Trans. Multimedia*, vol. 22, no. 8, pp. 2177–2190, Aug. 2020.
- [42] F. Xiong et al., "A2J: Anchor-to-joint regression network for 3D articulated pose estimation from a single depth image," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 793–802.
- [43] Y. Xu, J. Zhang, Q. Zhang, and D. Tao, "ViTPose: Simple vision transformer baselines for human pose estimation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 38571–38584.
- [44] D. Ye, Y. Xie, W. Chen, Z. Zhou, and H. Foroosh, "LPFormer: LiDAR pose estimation transformer with multi-task network," in *Proc. IEEE Int. Conf. on Robot. Automat.*, pp. 16432–16438, 2024.
- [45] M. Ye and R. Yang, "Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 2345–2352.
- [46] Y. Yoshiyasu, "Deformable mesh transformer for 3D human mesh recovery," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 17006–17015.
- [47] A. Zanfir et al., "HUM3DIL: Semi-supervised multi-modal 3D human pose estimation for autonomous driving," in *Proc. 6th Conf. Robot Learn.*, 2023, pp. 1114–1124.
- [48] H. Zhang, J. Cao, G. Lu, and W. Ouyang, and Z. Sun, "Learning 3D human shape and pose from dense body parts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2610–2627, May 2022.
- [49] M. Zhang et al., "A flexible multi-view multi-modal imaging system for outdoor scenes," in *Proc. 2022 Int. Conf. 3D Vis.*, 2022, pp. 322–331.
- [50] J. Zhen et al., "SMAP: Single-shot multi-person absolute 3D pose estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 550–566.
- [51] J. Zheng et al., "Multi-modal 3D human pose estimation with 2D weak supervision in autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 4478–4487.
- [52] S. Zou et al., "Human pose and shape estimation from single polarization images," *IEEE Trans. Multimedia*, vol. 25, pp. 3560–3572, 2023.
- [53] X. Zuo, S. Wang, Q. Sun, M. Gong, and L. Cheng, "Self-supervised 3D human mesh recovery from noisy point clouds," 2021, *arXiv:2107.07539*.



Bohao Fan received the B.S. degree in 2019 from the Department of Automation, Tsinghua University, Beijing, China, where he is currently working toward the Ph.D. degree with the Department of Automation. His research interests include human pose estimation and point cloud processing.



Wenzhao Zheng (Member, IEEE) received the B.S. degree from the Department of Physics and the Ph.D. degree from Department of Automation, Tsinghua University, Beijing, China, in 2018 and 2023, respectively. His research interests include unsupervised representation learning, vision-centric autonomous driving, and explainable artificial intelligence.



Jianjiang Feng (Member, IEEE) received the B.S. and Ph.D. degrees from the School of Telecommunication Engineering, Beijing University of Posts and Telecommunications, Beijing, China, in 2000 and 2007, respectively. He is currently an Associate Professor with the Department of Automation, Tsinghua University, Beijing. His research interests include fingerprint recognition and computer vision. He is an Associate Editor for IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY and IEEE TRANSACTIONS ON BIOMETRICS, BEHAVIOR, AND IDENTITY SCIENCE.



Jie Zhou (Fellow, IEEE) received B.S. and M.S. degrees from the Department of Mathematics, Nankai University, Tianjin, China, in 1990 and 1992, respectively, and the Ph.D. degree from the Institute of Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology, Wuhan, China, in 1995. Since 1997, he has been a Postdoctoral Fellow with the Department of Automation, Tsinghua University, Beijing, China, where he has been a Full Professor with the Department of Automation since 2003. His research interests include computer vision, pattern recognition, and image processing. He is an Associate Editor for IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE.