Dynamic Mesh Recovery from Partial Point Cloud Sequence

Hojun Jang[1] Minkwan Kim[1] Jinseok Bae[1] Young Min Kim[1,2]

[1] Dept. of Electrical and Computer Engineering, Seoul National University
[2] Interdisciplinary Program in Artificial Intelligence and INMC, Seoul National University
{j12040208, mkjjang3598, capoo95, youngmin.kim}@snu.ac.kr

Abstract

The exact 3D dynamics of the human body provides crucial evidence to analyze the consequences of the physical interaction between the body and the environment, which can eventually assist everyday activities in a wide range of applications. However, optimizing for 3D configurations from image observation requires a significant amount of computation, whereas real-world 3D measurements often suffer from noisy observation or complex occlusion. We resolve the challenge by learning a latent distribution representing strong temporal priors. We use a conditional variational autoencoder (CVAE) architecture with a transformer to train the motion priors with a large-scale motion dataset. Then our feature follower effectively aligns the feature spaces of noisy, partial observation with the necessary input for pre-trained motion priors, and quickly recovers a complete mesh sequence of motion. We demonstrate that the transformer-based autoencoder can collect necessary spatio-temporal correlations robust to various adversaries, such as missing temporal frames, or noisy observation under severe occlusion. Our framework is general and can be applied to recover the full 3D dynamics of other subjects with parametric representations.

1. Introduction

Understanding human motion is important for many real-world applications to assist humans. It has been a consistent interest of research, and has witnessed remarkable progress. Many previous works attempted to detect human poses by locating predefined joints from 2D images and fitting template meshes. 3D information has to be inferred as post-processing with either multi-view observations or by incorporating prior knowledge about the size of the body or objects. A large volume of works also find 3D poses from 3D data of marker-based motion captures, or gyroscopes, which require additional hardware attached to the body parts. On the other hand, point clouds are easy to obtain using a commodity depth camera or a LiDAR sensor observing the scene. By directly measuring the 3D locations of the parts, we can easily reason about body positions relative to surroundings and have advantages in inferring the consequences of human-object interaction or human-human interaction.

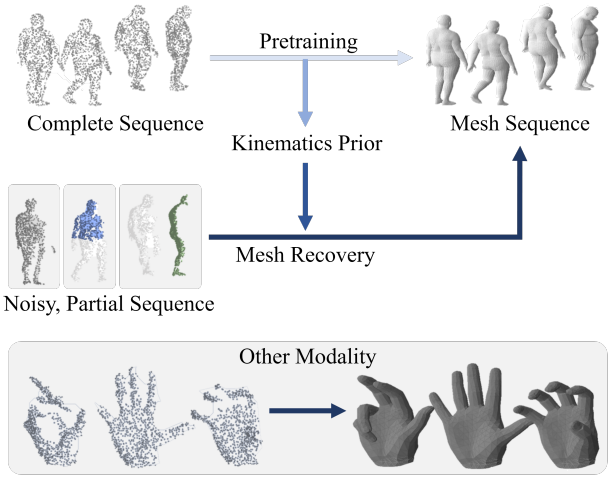

Figure 1. Overview of our approach. Our model firstly learns to recover mesh sequence from the complete point cloud sequence. Using this kinematics prior, we train our model in various other scenarios and recover the mesh sequence. Our model can also be trained on other input modality to generate its mesh sequence.

We propose a pipeline to obtain full 3D dynamic mesh from noisy, partial point cloud sequences. Real-world observations of motion are highly complex and suffer from occlusion by other objects or noises. Nonetheless, humans can easily infer the motion context of other humans. Not only does the physical connectivity of the skeleton structure define the range of possible motions for human body parts, but the motion semantics result in a rich correlation between temporal frames. To this end, we observe a sequence of point cloud measurements, instead of individual frames, and utilize strong kinematics information obtained from a large-scale motion data. Note that the overall pipeline is not bounded to human mesh, but can also be extended to other

subjects with kinematic structures.

We gain robustness against complex real-world challenges with a generative prior obtained from a large-scale motion sequences and a transformer to focus on reliable evidence. Given noisy partial measurements, there are many possible motions to explain the observations. Instead of regressing for a single deterministic pose of a given time step, we embrace the uncertainty by maintaining the distribution of latent space of motions with conditional variational autoencoder (CVAE). After the kinematic priors are obtained as a latent distribution, we can sample a latent vector and generate a plausible mesh output. We also employ the transformers to apply attention to meaningful observations while robust to unknown missing data. We demonstrate the superior performance of our full 3D motion recovery compared to other approaches focusing on single-time steps or deterministic methods.

Our contributions are summarized as follows:

- We demonstrate reusable 3D kinematic priors from large-scale motion datasets can provide strong structural semantics to recover dynamic 3D mesh in various scenarios.
- We show multi-modal motion prediction using variational frameworks and transformers, and demonstrate that both are critical to maintaining stable performance in challenging inputs.
- Our framework achieves superior performance over other existing methods, and can be generally applied for a diverse set of motions.

We expect the proposed method to provide an essential tool to capture and understand human motion in real-world scenarios and extend to practical applications to assist humans.

2. Related Work

In this paper, we present a two-stage learning scheme for dynamic mesh recovery from the sequential observation. Our proposed framework includes pre-training of the kinematics priors with a transformer from the massive amounts of motion capture data, and finally reuses learned priors to fully recover mesh sequence from noisy, partial observation. We review relevant works on human pose estimation techniques and the motion learning pipeline for the large-scale dataset.

Human pose estimation is a classical field in the computer vision community, and has experienced tremendous progress with recent studies on the data-driven approach. Several works [37, 15, 46, 22, 9] have extracted 2D skeletons from the image inputs in various ways. Previous works on the 2D pose estimation either take a top-down or bottom-up approach. The top-down approach starts from detecting the overall bounding box [8, 43], whereas the bottom-up approach first detects joints with keypoints [41] or heatmaps [2]. Some works remedy the limitation of 2D observation by incorporating multi-view settings [19] or depth images [50]. While such attempts improve the prediction quality especially in handling occlusions, they are insufficient to overcome the lack of 3D information.

Point cloud is a great option for the human pose estimator since it contains the spatial 3D information, and can be directly acquired from commodity depth sensors. Recently, some works proposed neural architectures [33, 34, 11] that are specialized in processing point cloud, and they have empirically proved their efficacy on perceiving shapes. Further researches have expanded the applications for the proposed perception modules via conducting classification [3], completion [45], and generation [44] of a 3D shape with the point cloud data. Shape of a soft body target, such as human, can be also be represented with a point cloud. Several works estimate skeletal poses directly on the captured point cloud of a human body [47, 25, 42, 1]. Other works extend from the skeleton and infer the full 3D body mesh [20, 24, 52, 4, 40] utilizing a parametric model [21, 30]. However, they only estimate the body pose of individual frames, ignoring the temporal aspect of the motion. Instead, our model incorporates structural priors as well as temporal correlations in sequential input and recover the full 3D mesh with a smooth motion.

By learning and incorporating pose priors of human, the estimated poses stay in the range of natural human motion, greatly enhancing the quality of results given challenging observations. Previous works obtain pose priors from real observations and constrain the pose estimation results with inverse kinematics [29, 5]. Similarly, motion capture dataset [23, 13, 16] can provide task-agnostic reference for human motion. Motion priors can also be obtained from demonstrations [35]. Additionally, some works focus on obtaining high-level semantics with text description [12, 31]. The resulting motion, however, can merely be regraded as a weak guidance as the training data is not perfect in terms of the consistency and the accuracy of labels. In contrast, our implicit kinematics prior is trained to capture extensive spatio-temporal information, and can be used to estimate poses in a variety of downstream scenarios.

3. Method

Given a sequence of point cloud measurements $\mathbf{P} = \{P_t\}_{t=1}^{T}$, we recover the dynamic sequences of full mesh from a set of estimated parameters. The mesh is recovered as we estimate the pose $\Theta = \{\theta_t\}_{t=1}^{T}$, $\theta_t \in \mathbb{R}^M$, the root position $\mathbf{x} \in \mathbb{R}^{T\times 3}$, and shape parameters β of a parametric model such that it best explains the input. For the human datasets, each parameter set is fitted into SMPL [21] to generate a full human mesh, whereas for the hand datasets,

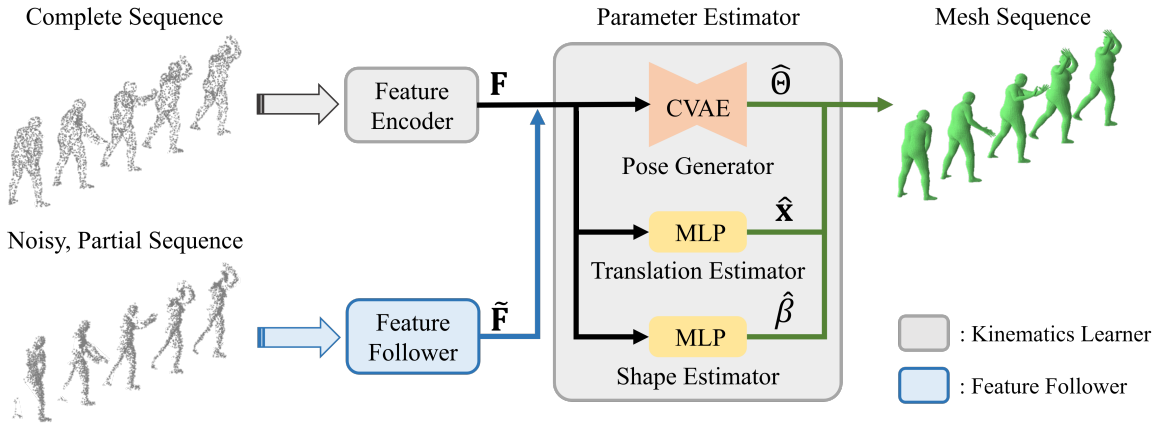

Figure 2. Overall pipeline of our method. The kinematics learner, the point cloud feature encoder followed by the parameter estimator, learns how to recover mesh from the complete point cloud sequence input. The feature follower then follows the encoding of the feature encoder in the kinematics learner. The learning of the feature encoding makes our model to effectively handle noisy, partial point cloud sequence inputs.

parameters are fitted into MANO [36] to reconstruct the human hand mesh. We mainly focus on obtaining the plausible pose parameters, which is then combined with the root position to generate joint positions. The joint positions are obtained from the joint regressor, $J_k(\theta, x) : \{\mathbb{R}^M, \mathbb{R}^3\} \rightarrow \mathbb{R}^3$, which combines the given information with the pre-defined kinematic structure of the parametric model.

The overall pipeline is described in Figure 2. We first obtain the motion prior by training the *kinematics learner* with a set of complete sequences, then quickly learn to extract the aligned feature with a *feature follower* for incomplete inputs. The kinematics learner establishes a rich latent space that can distinguish a large-scale motion dataset as described in Section 3.1. Then for a scarce, noisy input, we train a feature follower which replaces the feature encoder and extracts features aligned with the feature embedding of the kinematics learner (Section 3.2). The detailed architecture of the networks is provided in the supplementary material.

3.1. Kinematics Learner

The kinematics learner is trained with large-scale temporal datasets with ground truth parameters to obtain the strong motion prior. It is composed of a feature encoder and a parameter estimator. The feature encoder compresses the high-dimensional input of the point cloud sequence, and obtains an intermediate feature embedding $\mathbf{F} = \{F_t\}_{t=1}^T, F_t \in \mathbb{R}^{D_F}$. Subsequent parameter estimator can decode the features into pose, translation, and shape parameters, which can retrieve the motion sequence of 3D mesh. In particular, we adapt a CVAE architecture which observes the input feature $\mathbf{F}$ to estimate the pose parameters Θ. Once trained, the kinematics learner obtains a powerful embedding space that can capture various temporal configurations of a parametric mesh model. Then the feature follower can take an advantage of the intermediate latent space for limited data. Once the feature follower can map the input sequence into the feature space, we can transform observed sequences into the dynamic 3D mesh sequence with the pre-trained parameter estimator.

Feature Encoder The feature encoder is composed of a PointNet that extracts the initial features for individual frames, followed by a transformer network [39] that aggregates information in a temporal window. The input to the kinematics learner is a sequence of point clouds, composed of T frames of N 3D points, $\mathbf{P} = \{P_t\}_{t=1}^T$, $P_t \in \mathbb{R}^{N\times 3}$. A PointNet architecture [32] first embed point clouds for individual time steps, and retrieve PointNet features $\mathbf{P}' = \{P'_t\}_{t=1}^T$, $P'_t \in \mathbb{R}^{D_P}$. Then the individual PointNet features P'_t are inserted to separate transformer channels, where the temporal index t is subject to positional encoding to embed time stamp information of the feature. The transformer provides necessary attention to the time window and outputs aggregated feature $\mathbf{F}$. We jointly train the PointNet and the transformer encoder with the parameter estimator with the ground-truth pairs of complete point cloud input and the mesh parameter output.

In addition to the losses combined with the final parameter estimation, we define an auxiliary task to guarantee that the intermediate feature $\mathbf{F}$ contains sufficient motion semantics. We build a small multilayer perceptron (MLP) to ensure that the extracted features can estimate the ground-truth pose parameters $\Theta = \{\theta_t\}_{t=1}^T$,

$$\Theta^{\text{aux}} = \text{MLP}^{\text{aux}}(\mathbf{F}). \tag{1}$$

Then we can define the following loss term:

$$\mathcal{L}_\theta^{\text{aux}} = \frac{1}{T}\sum_t \|\theta_t^{\text{aux}} - \theta_t\|_2^2 \tag{2}$$

where θ is the ground truth pose parameter and θ^{aux} is the parameter estimate.

Parameter Estimator The parameter estimator finds the necessary parameters for the mesh recovery from the encoded feature $\mathbf{F}$. Specifically, we have to find the pose parameters $\Theta = \{\theta_t\}_{t=1}^{T}, \theta_t \in \mathbb{R}^M$, root translations $\mathbf{x} = \{x_t\}_{t=1}^{T}, x_t \in \mathbb{R}^3$, and the shape parameter $\beta \in \mathbb{R}^S$. Note that the shape parameter β is a constant for a given temporal sequence assuming we are observing the same subject.

We use a simple deterministic formulation to estimate the root translation $\mathbf{x}$ and the shape β. Specifically, we use two neural networks composed of MLPs to regress for root positions $\text{MLP}^x(\mathbf{F}) = \mathbf{x}$ and the shape parameter $\text{MLP}^\beta(\mathbf{F}) = \beta$, respectively.

In contrast to the simple deterministic model for estimating $\mathbf{x}$ and β, we use a generative model of CVAE to find the distribution of possible pose parameters Θ. When the input data is noisy or severely occluded, there can be multiple plausible sequences of motion, and the generative model stabilizes the estimation in such cases. Our CVAE architecture is composed of a prior distribution $p_\psi(\mathbf{z}|\mathbf{F})$ and a posterior distribution $q_\phi(\mathbf{z}|\Theta, \mathbf{F})$. $\mathbf{z} = \{z_t\}_{t=1}^{T}, z_t \in \mathbb{R}^{D_z}$ is a latent variable that encodes motion features, and it is sampled for generation. The sampled latent variable is then decoded to a set of pose parameters Θ using a transformer-based pose decoder.

Training for the posterior distribution q_ϕ forms a distribution of latent variable $\mathbf{z}$ that captures the motion information. The loss term used to generate plausible pose parameters is written below:

$$\mathcal{L}_\theta = \frac{1}{T}\sum_t \|\hat{\theta}_t - \theta_t\|_2^2 \tag{3}$$

where $\hat{\theta}$ is the pose reconstructed from the decoder.

We further encourage the estimation to be similar to the ground-truth by imposing additional losses. We add two losses to the joint positions estimated from the pose parameters θ and the root positions x. First, we directly compare the joint positions with the joint reconstruction loss:

$$\mathcal{L}_J = \frac{1}{TK}\sum_{t,k} \|J_k(\hat{\theta}_t, \hat{x}_t) - J_k(\theta_t, x_t)\|_2^2 \tag{4}$$

where $\hat{\theta}$ and $\hat{x}$ are estimated pose and translation, and $J_k(\cdot)$ is a joint regressor which outputs the k-th joint position from pose and translation parameters. The second loss is the volume fitting loss introduced in [1] and holds the joint locations near the input point cloud:

$$\mathcal{L}_{vol} = \frac{1}{T}\sum_t \frac{1}{|P_t|}\sum_{p_t \in P_t} \min_k \|p_t - J_k(\hat{\theta}_t, \hat{x}_t)\|_2^2. \tag{5}$$

It is basically an one-directional Chamfer Distance [7], measured from the joints to the point cloud. The volume fitting loss was initially proposed to uniformly spared the keypoints within the occupied 3D voxels [1], but our loss substitutes the estimated joints and the point cloud measurements. Lastly, we design the shape loss to maintain a realistic shape parameter:

$$\mathcal{L}_\beta = \|\hat{\beta} - \beta\|_2^2, \tag{6}$$

where $\hat{\beta}$ is the estimated shape parameter and β is the ground truth shape parameter.

While the posterior distribution q_ϕ is trained with the loss terms above ($\mathcal{L}_\theta$, $\mathcal{L}_J$, and $\mathcal{L}_{vol}$), the prior distribution p_ψ is encouraged to follow the distribution of q_ϕ with the KL divergence term which matches the observation with the evidence lower bound (ELBO) [18]:

$$\mathcal{L}_{KL} = \frac{1}{T}\sum_t D_{KL}(q_\phi(z_t|\theta_t, F_t)\|p_\psi(z_t, F_t)). \tag{7}$$

3.2. Feature Follower

Once the reusable kinematic priors are obtained, the feature follower brings a scarce, noisy point cloud sequences to the input feature space of the kinematic learner. Then the parameter estimator, trained from the kinematics learner, can stably recover the dynamic mesh. The feature follower has the same architecture as the feature encoder of the kinematics learner, namely the feature extractor of PointNet [32] followed by the transformer encoder. We train the network with the loss to match the features between the kinematics learner and the feature follower:

$$\mathcal{L}_F = \frac{1}{T}\sum_t \|\tilde{F}_t - F_t\|_2^2, \tag{8}$$

where $\tilde{\mathbf{F}} = \{\tilde{F}_t\}_{t=1}^{T}$, $\tilde{F}_t \in \mathbb{R}^{D_F}$ represents the feature encoded from the noisy, partial point cloud sequence while $\mathbf{F}$ is extracted from the full point cloud sequence. The feature follower also employs the additional loss terms used to train the kinematics learner ($\mathcal{L}_\theta^{\text{aux}}$, $\mathcal{L}_\theta$, $\mathcal{L}_J$, and $\mathcal{L}_\beta$). $\mathcal{L}_{vol}$ is excluded, since the estimated joint locations cannot always be located near the input when the input is especially a single-view or partial point cloud.

Note that we only train the substitute of the encoder part of the kinematics learner for diverse situations of input sequence. Once we obtain the feature embedding of the new input sequences, we utilize the pre-trained parameter estimator to recover the full 3D mesh. Moreover, by applying random temporal masks in the transformer part of the encoder, the entire process becomes robust to temporal adversaries with missing frames. Along with the variational inference, the transformer serves a crucial role for stable performance, which is further verified in the experiment.

4. Experiments

In this section, we demonstrate how the feature follower in Section 3.2 utilizes the reusable kinematic prior in Sec-

tion 3.1 to enhance the performance in noisy, partial input sequences. We firstly demonstrate the performance of our pipeline over the baselines on human datasets (Section 4.1). We further show that our framework is generalizable to other types of parametric mesh models by sharing the results with a hand model, which is highly challenging due to severe self-occlusion (Section 4.2).

Our method is mainly implemented using Pytorch Lightning package [6], and is accelerated with RTX 4090 GPU for the kinematics learner and RTX 3090 GPU for the feature follower, respectively. In all our experiments the number of the points used is 1,024 and the total length of the sequence is set to $T = 40$ with the frame rate of 10 fps for the human dataset, and 5 fps for the hand dataset. The train-test split for each experiments and additional hyperparameter setup are included in the supplementary material. We report the performance of each model using pose error, joint error, and vertex error. The pose and joint errors are the mean L2 distances in angle and location, respectively, between the estimated joints and the ground-truth. The vertex error is a mean per-vertex L2 distance between the estimated mesh vertex and the ground-truth. All the errors are evaluated in the parameter space (SMPL [21] for human, and MANO [36] for hand) and averaged along the time step.

Baselines We compare the performance of dynamic mesh recovery against four baselines. Two methods are based on point cloud registration and the other two recover full mesh based on the parametric model.

3D-CODED [10] aims to find the correspondences between reference and target shapes. An autoencoder framework with an additional template mesh is proposed to recover shapes from point cloud inputs. It first uses a neural network to deform the template mesh to fit the input point cloud, followed by an extra step of local optimization to minimize the Chamfer Distance [7]. We report both results with and without the additional optimization process. Unlike our method, 3D-CODED does not originally utilize severely partial point cloud so that the method is unable to show its full capability in scenarios for our experiment. Besides, SyNoRiM [14] registers multiple point clouds by synchronizing the functional maps defined on the point clouds. We train the network to estimate the flow of the point clouds from the template mesh to the partial point clouds. As our datasets do not include correspondences between input points and points on a template, we train both baselines using the Chamfer Distance with additional Laplacian and edge regularization terms.

VoteHMR [20] resolves the challenges from occlusion and measurement noises of single-view point cloud measurement with additional information. Specifically, it observes part segmentation of input point cloud and classify them into different joints. In addition to the part segments, it also assumes that the root translation is known, while both pieces of information are not necessary in our approach. Meanwhile, Zuo *et al*. [52] reconstruct the mesh surface of the human body based on optimization. From the point cloud input, they first estimate the parameters with a neural network, and subsequently refine the fitting with probabilistic self-supervised loss functions. While this approach is robust to outliers in the training dataset unlike usual learning-based methods, suggested two-step approach is far from real-time implementation due to a considerable amount of computation in optimization steps. Here, we compare our method to cases with and without the additional fitting steps. Note that VoteHMR and Zuo *et al*. separately estimate the pose parameters for individual frames, whereas we concurrently regress parameters for a temporal sequence. To the best of our knowledge, there is no baseline available that fits dynamic mesh to a sequence of point cloud frames.

4.1. Human Motion

We use AMASS dataset [23] as the large-scale dataset to learn priors on human motion information. AMASS dataset contains more than 40 hours of motion sequences and spanning over 300 subjects, with the full SMPL [21] parameters fitted from motion capture data. We train the kinematics learner network with the dataset except CMU dataset [17].

After training the kinematics learner, we train our feature follower network for various motion scenarios. We synthetically generate various corrupted inputs to test the performance of the feature follower. We follow the process of SURREAL dataset [38] to emulate the point cloud obtained from a depth camera. The SURREAL dataset makes the depth sequence by projecting mesh model into a background, and employs the same setup of SMPL parameters as the CMU dataset [17]. We similarly project the mesh, generated by fitting SMPL parameters of CMU dataset [17], to obtain depth image sequence. The depth image is then transformed into a point cloud by adapting the process introduced in [20]. Therefore, CMU dataset is excluded in training the kinematics learner, and adapted to generate realistic noisy sequences to train the feature follower.

Synthetic Data with Emulated Noise The results of our method and the baselines on synthetic human motion dataset are shown in the Table 1. Note that we only report the vertex displacement error for 3D-CODED [10] and SyNoRiM [14] as they do not estimate the SMPL parameters. Our model shows the best performance among the non-optimization baselines and even comparable to the baseline Zuo *et al*. [52], which performs additional optimization for mesh recovery. Zuo *et al*. and 3D-CODED with the fitting process take about 10 minutes to fit a full sequence using

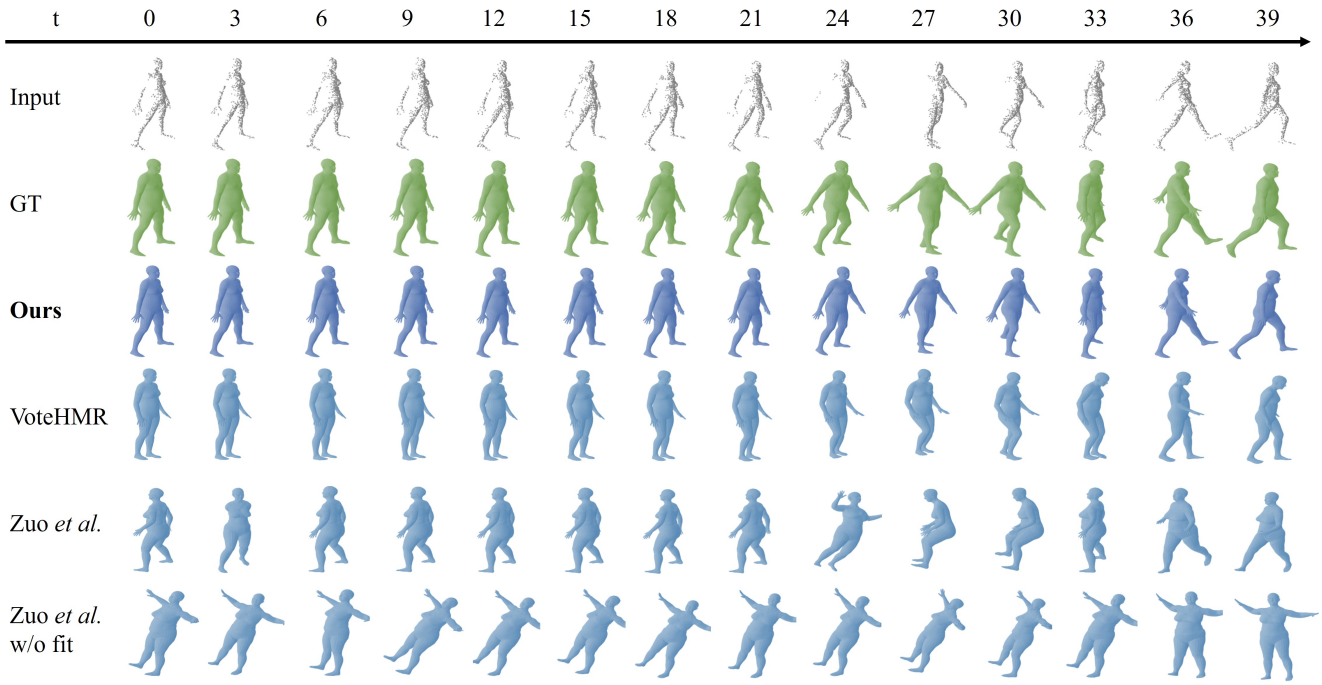

Figure 3. Qualitative results on synthetic human motion dataset. The first row and the second row show the input point cloud sequence and the corresponding ground-truth mesh sequence. The rows below are the mesh recovery result of our model and the baselines.

Method	Optim.	Pose	Joint	Vertex
3D-CODED [10]	○	-	-	0.1092
Zuo *et al*. [52]	○	**0.3354**	0.0824	0.1109
3D-CODED w/o fit	×	-	-	0.1154
SyNoRiM [14]	×	-	-	0.0219
Zuo *et al*. w/o fit	×	0.4095	0.1924	0.2625
VoteHMR [20]	×	0.4142	0.0127	0.0258
Ours	×	**0.3545**	**0.0109**	**0.0132**

Table 1. Mesh reconstruction performance of our model and the baselines on synthetic human motion dataset.

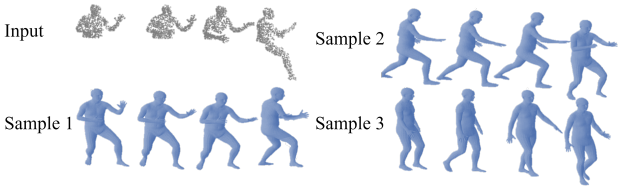

Figure 4. The example of our model producing multiple outputs for a severely partial point cloud sequence. The input point cloud sequence lacks body information and our model generates multiple plausible sequences.

a single batch. All other methods without the optimization, including ours, take less than a second to estimate the parameters for a full sequence. Figure 3 shows the qualitative comparison for methods that directly estimate the SMPL [21] parameters. The methods based on point cloud registration are not visualized as they ruin the topology of the output mesh by mixing the order of vertices after matching. Baselines estimate the parameters for individual frames of input, and result in unnatural jittering motion within the sequence. On the other hand, our method correctly captures temporal correlation and reconstruct meshes with smooth motion. Such phenomenon is better observable in the supplementary video.

Spatially Partial Sequence We also present results on training a feature follower with an input sequence with largely occluded region. Because our pose generator is composed of a CVAE architecture, we can sample the latent vector from the learned distribution to generate multiple plausible mesh outputs. Figure 4 shows the output generated from multiple samples, each sampled from 1σ, 2σ, and 3σ, respectively. With the help of variational framework, our pipeline can generate diverse output sequences given challenging ambiguous input.

Temporally Partial Sequence We further present results handling temporally missing data. The scenario reflects the cases where data is sporadically unreliable, for example, with a network issue dropping intermediate frames. To cope with this situation, we take advantage of the transformer networks used in our feature encoder and decoder. We manipulate the key padding mask in the transformer network to hold the transformer attention from relying on the blocked time step. In the training phase, we set the key padding

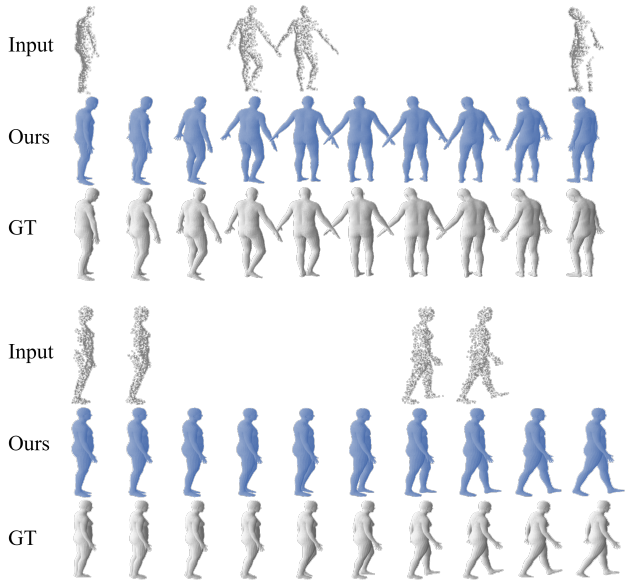

Figure 5. The example of our model showing results on temporally partial sequence inputs. Our model successfully reconstructs the mesh even when there is no data along time.

Method	Optim.	Chamfer Dist.
3D-CODED [10]	○	0.0101
Zuo *et al*. [52]	○	**0.0014**
3D-CODED w/o fit	×	0.0559
SyNoRiM [14]	×	0.0042
Zuo *et al*. w/o fit	×	0.0092
VoteHMR [20]	×	0.0051
Ours	×	**0.0027**

Table 2. Mean Chamfer Distance of the methods on Berkeley MHAD dataset [27]. The Chamfer Distance between the recovered mesh and the input point cloud is averaged through the time.

mask to block a sequence with a random ratio with an upper bound. This way, the encoder and decoder learn how to handle the sequence with missing frames at unknown time steps. Figure 5 shows the mesh recovery result tested in a temporally partial sequence. Our model effectively recovers the complete sequence including the erased time steps.

Real Data We additionally test our model on real human motion dataset, Berkeley MHAD [27]. Berkely MHAD dataset, obtained from a Microsoft Kinect, is a depth data of about 660 action sequences performed by 12 actors. We transform the depth data into a point cloud sequence and use it as an input of our model. Note that the dataset captures the entire environment where a human is present, and we roughly crop the points near the actor position using a bounding box. We retrieve the cropping bounding boxes from mocap sensor positions, which the dataset provides along with the depth sequences.

Method	Pose	Joint	Vertex
VoteHMR-M [20]	0.3036	**0.0101**	0.0091
Ours	**0.2353**	**0.0101**	**0.0088**

Table 3. Evaluation results of our model and VoteHMR-M on partial hand sequence. VoteHMR-M is a modified version of VoteHMR [20] which can handle human hand dataset.

We test the generalization of our model on Berkeley MHAD dataset [27] and compare against the baselines, as shown in Table 2. We report the Chamfer Distance [7] between the estimated human mesh vertices and the input point cloud averaged along the time period. Even though the input point cloud is extremely noisy and the network is not fine-tuned to the noisy real data, our model recovers the mesh sequence comparable to the optimization-based method as shown in Figure 6. The error value of the model Zuo *et al*. [52] is the smallest among all the methods. However, as shown in the figure, results from Zuo *et al*. show inconsistent body directions through the time step.

Our method rarely fails to recover the mesh sequence properly in extreme cases, such as a sequence with a chair to sit on, or when there is a severe noise in the sequence so that the shape parameter is estimated to generate a fatter mesh output.

4.2. Hand Motion

To evaluate the performance on other types of parametric mesh model, we use the hand model with two datasets. HanCo [49, 48] and InterHand2.6M [26] are hand motion datasets which fit MANO [36] parameters to the hand motion capture data. For training the kinematics learner, we use full point cloud input sequence to learn rich kinematic priors about the hand motion. In the feature follower training phase, we generate single-view point cloud sequence similar to the human partial point cloud dataset. The train-test splits of the kinematics learner and the feature follower is given in the supplementary material.

VoteHMR [20] is modified to be trained with the MANO [36] parameters and made to output hand mesh. We name the VoteHMR model modified for the MANO parameters as VoteHMR-M. Table 3 summarizes the reconstruction results, which indicates that our method has less error values than VoteHMR-M in pose and vertex displacement. Figure 7 shows the qualitative results.

4.3. Ablation Study

In this section, we discuss the effectiveness of our method. We first analyze the efficacy of $\mathcal{L}_{\theta}^{\text{aux}}$ and $\mathcal{L}_{vol}$ used when training the kinematics learner. Table 4 shows the results of the kinematics learner each trained with or without $\mathcal{L}_{\theta}^{\text{aux}}$ and $\mathcal{L}_{vol}$, respectively. To shorten the training time of the experiments, we randomly sampled one-tenth of the

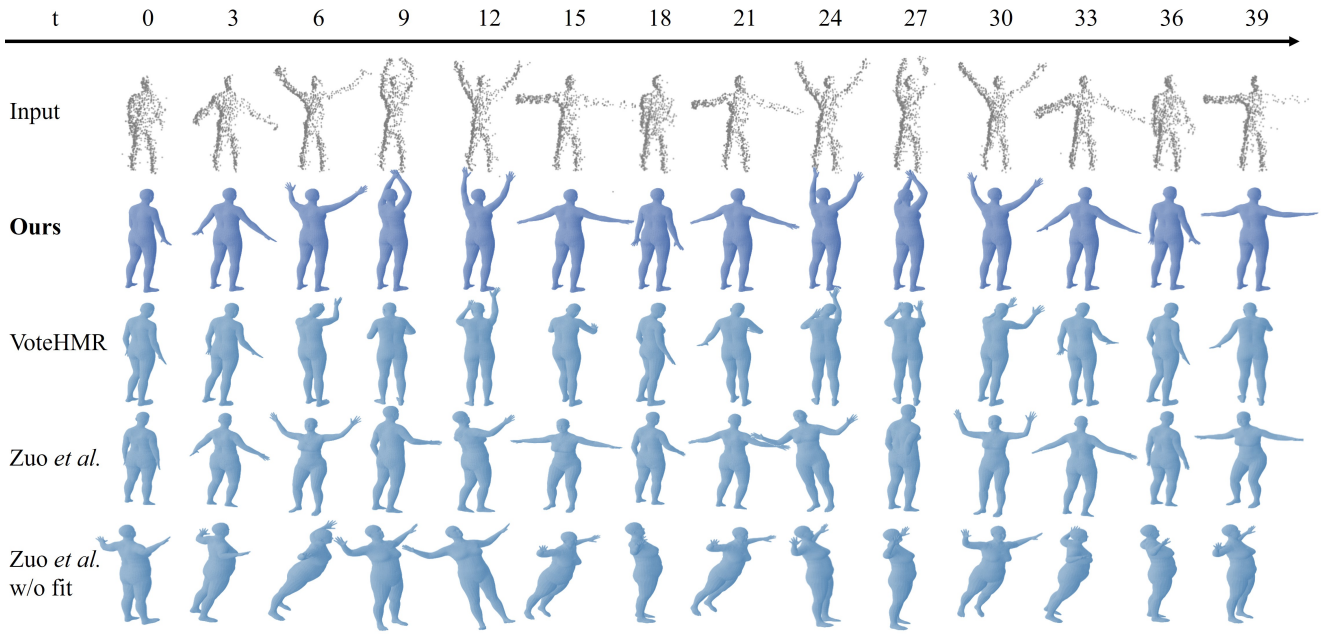

Figure 6. Qualitative results on Berkeley MHAD dataset [27]. While the results from our method and VoteHMR [20] show certain body direction, the result from Zuo *et al*. [52] keeps flipping.

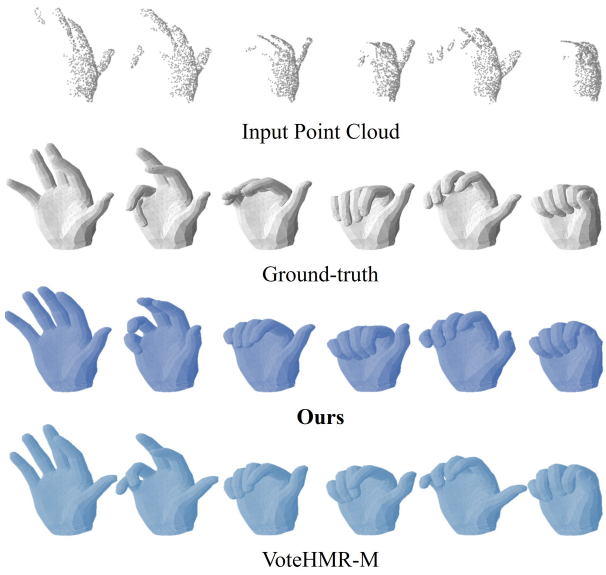

Figure 7. Qualitative results on hand motion dataset. The top two rows show the input sequence of the point cloud and its corresponding ground-truth mesh. The third row shows the recovery result of our model and the last shows the results from VoteHMR-M, VoteHMR [20] modified for human hand.

Module	Method	Pose	Joint	Vertex
Kinematics Learner	Ours w/o $\mathcal{L}_\theta^{\text{aux}}$, $\mathcal{L}_{vol}$	0.3856	0.0928	0.1539
	Ours w/o $\mathcal{L}_\theta^{\text{aux}}$	0.3755	0.0900	0.1217
	Ours w/o $\mathcal{L}_{vol}$	0.3146	0.0573	0.0309
	Ours	**0.3064**	**0.0504**	**0.0215**
Feature Follower	Ours *Direct*	0.7293	0.0150	0.1018
	Ours w/o $\mathcal{L}_F$	0.3556	0.0138	0.0166
	Ours *Small*	0.3567	0.0157	0.0190
	Ours	**0.3545**	**0.0109**	**0.0132**

Table 4. Additional experiments on our model for the kinematics learner and the feature follower. Ours *Direct* is the kinematics learner trained directly to the partial point cloud. Ours *Small* refers to the model with the smaller feature follower capacity.

AMASS [23] training data to train the kinematics learner. The results show that the loss terms introduced to train the kinematics learner were valid.

We then substantiate our design choice of using the feature follower and the feature following loss $\mathcal{L}_F$. Note the experiments are conducted on the synthetic noisy human point cloud sequence. We test whether the pretraining scheme truly improves the motion tracking quality. We report the result of the kinematics learner trained directly to the partial point cloud sequence, which is end-to-end learning from partial point cloud to full mesh recovery. Moreover, we ablate the feature following loss $\mathcal{L}_F$ to test the effectiveness of following the feature encoded from the pretrained kinematics learner. Table 4 shows that the pretraining scheme and the introduced feature following loss were both effective. This implies that our pretraining strategy helps learning qualified feature space for the damaged point cloud sequence.

Additionally, we show whether a higher capacity of the feature follower network performs better or not. Here, we note that the feature follower network consists of PointNet [32] followed by the transformer encoder network as

described in Section 3.2. We designed the smaller model to have an equal number of parameters for the PointNet structure, while lower the capacity of the transformer encoder part to be quartered. As shown in Table 4, we found that higher capacity model outperforms smaller one, which indicates our method fully benefits the large model to learn appropriate feature encoding.

Future works In this work, we showed that our model can be applied to different kinds of parametric mesh model. Since our method uses no subject-specific terms for handling the input sequence, we expect our model to be applicable to other kinds of parametric mesh models, such as tetrapods [51] or other human models [28].

We notice several possible future extensions to enhance the practicality of the work. Our model fails to generate proper mesh reconstruction when there exists some point clouds in the floor or other objects. Advanced masking modules to filter out those undesirable points may further improve prediction of our model. Furthermore, we might design a new structure that has better robustness in noise handling. Lastly, our kinematics prior is reused only for the partial point cloud input. However, we think that our learned kinematics prior could be applied to a RGB-D input. By changing the feature follower module to handle additional RGB sequence will make the model capable to recover mesh sequence for the RGB-D sequence, thanks to the feature following loss.

5. Conclusion

We presented a method to fully recover mesh sequence from a noisy, partial point cloud sequence utilizing the pretrained kinematics prior. We use large-scale motion dataset to capture diverse movements of the subject. From the pretrained kinematics, we train our model to reconstruct the mesh sequence from the noisy, partial sequence. Our method benefits from the variational frameworks and transformers and shows strong reconstruction performance on partial sequences. CVAE architecture allows our model to generate multiple plausible pose parameters and shows an advantage especially on handling spatially partial point cloud sequence. Moreover, handling sequences with several empty data is available by manipulating the attention in the transformer network. This way, our model shows the best result among non-optimization baselines and even shows comparable results against the optimization-based baseline. We expect our model to go forward to take various kinds of input data in the future and further show its performance on many other applications.

Acknowledgments This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2023-00218601), Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) [NO.2021-0-01343, Artificial Intelligence Graduate School Program (Seoul National University)], and Creative-Pioneering Researchers Program through Seoul National University. Young Min Kim is the corresponding author.

References

[1] Jinseok Bae, Hojun Jang, Cheol-Hui Min, Hyungun Choi, and Young Min Kim. Neural marionette: Unsupervised learning of motion skeleton and latent dynamics from volumetric video. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(1):86–94, Jun. 2022. 2, 4

[2] Vasileios Belagiannis and Andrew Zisserman. Recurrent human pose estimation. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 468–475. IEEE, 2017. 2

[3] Yizhak Ben-Shabat, Michael Lindenbaum, and Anath Fischer. 3dmfv: Three-dimensional point cloud classification in real-time using convolutional neural networks. *IEEE Robotics and Automation Letters*, 3(4):3145–3152, 2018. 2

[4] Bharat Lal Bhatnagar, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Combining implicit function learning and parametric models for 3d human reconstruction. In *European Conference on Computer Vision (ECCV)*. Springer, August 2020. 2

[5] Andrey Davydov, Anastasia Remizova, Victor Constantin, Sina Honari, Mathieu Salzmann, and Pascal Fua. Adversarial parametric pose prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10997–11005, 2022. 2

[6] William Falcon and The PyTorch Lightning team. PyTorch Lightning, 3 2019. 5

[7] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 605–613, 2017. 4, 5, 7

[8] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 2334–2343, 2017. 2

[9] Zigang Geng, Ke Sun, Bin Xiao, Zhaoxiang Zhang, and Jingdong Wang. Bottom-up human pose estimation via disentangled keypoint regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14676–14686, 2021. 2

[10] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. 3d-coded: 3d correspondences by deep deformation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 5, 6, 7

[11] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *Proceedings of*

the IEEE conference on computer vision and pattern recognition, pages 216–224, 2018. 2

[12] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5152–5161, 2022. 2

[13] Félix G Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. Robust motion in-betweening. *ACM Transactions on Graphics (TOG)*, 39(4):60–1, 2020. 2

[14] Jiahui Huang, Tolga Birdal, Zan Gojcic, Leonidas J Guibas, and Shi-Min Hu. Multiway non-rigid point cloud registration via learned functional map synchronization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 5, 6, 7

[15] Junjie Huang, Zheng Zhu, Feng Guo, and Guan Huang. The devil is in the details: Delving into unbiased data processing for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5700–5709, 2020. 2

[16] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, jul 2014. 2

[17] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *The IEEE International Conference on Computer Vision (ICCV)*, 2015. 5

[18] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 4

[19] Muhammed Kocabas, Salih Karagoz, and Emre Akbas. Self-supervised learning of 3d human pose using multi-view geometry. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1077–1086, 2019. 2

[20] Guanze Liu, Yu Rong, and Lu Sheng. Votehmr: Occlusion-aware voting network for robust 3d human mesh recovery from partial point clouds. In *Proceedings of the 29th ACM International Conference on Multimedia*, MM '21, page 955–964, New York, NY, USA, 2021. Association for Computing Machinery. 2, 5, 6, 7, 8

[21] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015. 2, 5, 6

[22] Zhengxiong Luo, Zhicheng Wang, Yan Huang, Liang Wang, Tieniu Tan, and Erjin Zhou. Rethinking the heatmap regression for bottom-up human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13264–13273, 2021. 2

[23] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision (ICCV)*, pages 5442–5451, Oct. 2019. 2, 5, 8

[24] R. Marin, S. Melzi, E. Rodolà, and U. Castellani. Farm: Functional automatic registration method for 3d human bodies. *Computer Graphics Forum*, 39(1):160–173, 2020. 2

[25] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[26] Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. Interhand2.6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image. In *European Conference on Computer Vision (ECCV)*, 2020. 7

[27] Ferda Ofli, Rizwan Ahmed Chaudhry, Gregorij Kurillo, René Vidal, and Ruzena Bajcsy. Berkeley mhad: A comprehensive multimodal human action database. *2013 IEEE Workshop on Applications of Computer Vision (WACV)*, pages 53–60, 2013. 7, 8

[28] Ahmed A A Osman, Timo Bolkart, and Michael J. Black. STAR: A sparse trained articulated human body regressor. In *European Conference on Computer Vision (ECCV)*, pages 598–613, 2020. 9

[29] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10975–10985, 2019. 2

[30] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[31] Abhinanda R Punnakkal, Arjun Chandrasekaran, Nikos Athanasiou, Alejandra Quiros-Ramirez, and Michael J Black. Babel: bodies, action and behavior with english labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 722–731, 2021. 2

[32] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *arXiv preprint arXiv:1612.00593*, 2016. 3, 4, 8

[33] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2

[34] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 2

[35] Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J. Guibas. Humor: 3d human motion model for robust pose estimation. In *International Conference on Computer Vision (ICCV)*, 2021. 2

[36] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), Nov. 2017. 3, 5, 7

[37] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5693–5703, 2019. 2

[38] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 5

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 3

[40] Shaofei Wang, Andreas Geiger, and Siyu Tang. Locally aware piecewise transformation fields for 3d human mesh registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7639–7648, June 2021. 2

[41] Fangyun Wei, Xiao Sun, Hongyang Li, Jingdong Wang, and Stephen Lin. Point-set anchors for object detection, instance segmentation and pose estimation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pages 527–544. Springer, 2020. 2

[42] Guoqiang Wei, Cuiling Lan, Wenjun Zeng, and Zhibo Chen. View invariant 3d human pose estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(12):4601–4610, 2020. 2

[43] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 466–481, 2018. 2

[44] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4541–4550, 2019. 2

[45] Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. Pointr: Diverse point cloud completion with geometry-aware transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12498–12507, 2021. 2

[46] Feng Zhang, Xiatian Zhu, Hanbin Dai, Mao Ye, and Ce Zhu. Distribution-aware coordinate representation for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7093–7102, 2020. 2

[47] Yufan Zhou, Haiwei Dong, and Abdulmotaleb El Saddik. Learning to estimate 3d human pose from point cloud. *IEEE Sensors Journal*, 20(20):12334–12342, 2020. 2

[48] Christian Zimmermann, Max Argus, and Thomas Brox. Contrastive representation learning for hand shape estimation. *arXiv preprint arXiv:2106.04324*, 2021. 7

[49] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas Brox. Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 813–822, 2019. 7

[50] Christian Zimmermann, Tim Welschehold, Christian Dornhege, Wolfram Burgard, and Thomas Brox. 3d human pose estimation in rgbd images for robotic task learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1986–1992. IEEE, 2018. 2

[51] Silvia Zuffi, Angjoo Kanazawa, David Jacobs, and Michael J. Black. 3D menagerie: Modeling the 3D shape and pose of animals. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 9

[52] Xinxin Zuo, Sen Wang, Minglun Gong, and Li Cheng. Unsupervised 3d human mesh recovery from noisy point clouds. *CoRR*, abs/2107.07539, 2021. 2, 5, 6, 7, 8