# AniDynRecon: Animatable 4D Dynamics Reconstruction from Sparse Point Clouds Supplementary Document

**Anonymous Author(s)**
Affiliation
Address
email

## A Notations

In this paper, we use notations summarized in Tab. S1. $K$ denotes the number of joints, and $N$ represents the number of surface points. $M$ refers to the number of components used for the Gaussian Mixture Model (GMM). $x$ indicates the position of a surface point, while $p$ represents the position of a joint. $\Delta p$ is the delta position between a joint at the frame of rest pose and another frame. $f$ stands for a feature vector of rest pose encoded by PointNet++ Qi et al. (2017). $\hat{w}$ is the output of the joint weight MLP, and $\tilde{w}$ represents the joint weight. $w$ denotes the skinning weight. $d$ is the offset vector from a joint to a surface point, and $|d|$ is the distance between them. $R$ represents the rotation matrix of a joint, and $r^t$ is the rotation of a joint represented as lie algebra. $\mathcal{S}$ stands for shape tokens. $O$ denotes the occupancy value. $\bar{V}$ represents the vertices of a mesh, and $\bar{F}$ refers to the faces of a mesh.

| Notation | Type | Definition |
|---|---|---|
| $K$ | scalar | number of joints |
| $N$ | scalar | number of points |
| $M$ | scalar | number of components in the GMM |
| $x$ | vector | position of point |
| $p$ | vector | position of joint |
| $\Delta p$ | vector | delta position of joint |
| $f$ | vector | feature vector |
| $\hat{w}$ | scalar | output of joint weight MLP |
| $\tilde{w}$ | scalar | joint weight |
| $w$ | scalar | skinning weight |
| $d$ | vector | offset from joint to surface point |
| $\|d\|$ | scalar | distance from joint to surface point |
| $R$ | matrix | rotation matrix of joint |
| $r^t$ | vector | rotation of a joint at time $t$ |
| $\mathcal{S}$ | vector | shape token |
| $O$ | scalar | occupancy value |
| $\bar{V}$ | vector set | vertices of mesh |
| $\bar{F}$ | vector set | faces of mesh |

Table S1: Symbols and notations used in this paper.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

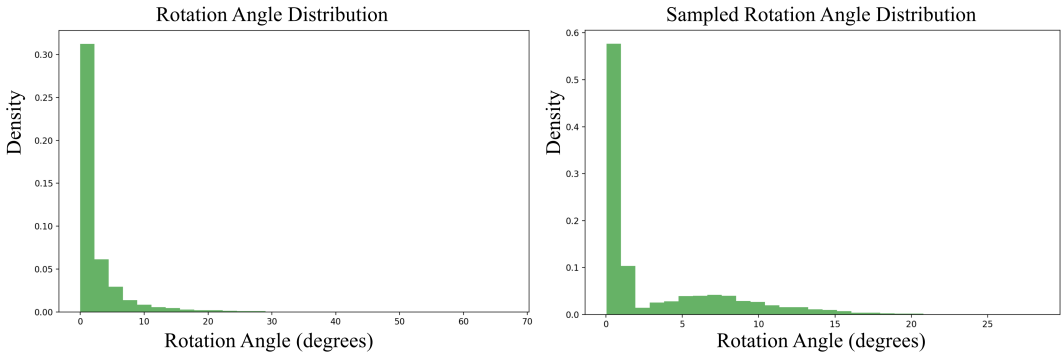


Figure S2: **(Left)**. Rotation angle distribution in original training data. **(Right)**. Sampled rotation angle distribution using GMM.

# B Implementation Details

## B.1 Final Training Objective

Our model is trained end-to-end, obtaining high quality when joint position and skinning weight field are jointly trained without any supervision under ground-truth. The total training objective of our network is:

$$\mathcal{L} = \lambda_{\text{locality}}\mathcal{L}_{\text{locality}} + \lambda_{\text{dynamics}}\mathcal{L}_{\text{dynamics}} + \lambda_{\text{joints\_inside}}\mathcal{L}_{\text{joints\_inside}} + \lambda_{\text{centrality}}\mathcal{L}_{\text{centrality}} + \lambda_{\text{shape}}\mathcal{L}_{\text{shape}} \quad (1)$$

We set the hyperparameters as follows: $\lambda_{\text{locality}} = 0.06, \lambda_{\text{dynamics}} = 0.001, \lambda_{\text{joints\_inside}} = 0.01, \lambda_{\text{centrality}} = 3.0, \lambda_{\text{shape}} = 1.0$. These values were determined based on the reconstructed IoU metric in our early experiments on partial datasets. In the experiments, we only adjusted the value of one weight at a time to determine these weights so that the network converges stably. Note that the Chamfer distance and the correspondence error reported in our paper, as well as Lei and Daniilidis (2022), are multiplied by 10.

## B.2 Details of Skinning Weight Decoder

## B.3 Network Settings

We design our skinning weight decoder $\mathcal{W}$ based on the observation that cross-attention between joints and vertices is similar to the blending weights in LBS. The architecture of the skinning weight decoder is depicted in Fig. S1. Surface points are first sampled via furthest point sampling (FPS). Then, positional embeddings (PE) of these points and subsampled points are fed into a cross-attention block to produce shape tokens $\mathcal{S}$. Subsequently, a self-attention block encodes the PE of joints and shape tokens. Then, another cross-attention block computes skinning weights using the encoded features as key and value, and the PE of surface points as query.

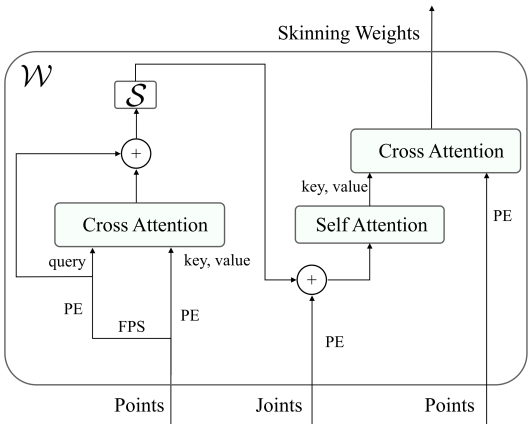


Figure S1: Architecture of our skinning weight decoder.

We leverage the 8-th frame of the input sequence as rest pose to reconstruct the shape of object. We follow the setting of Motion2Vecsets Cao et al. (2024): the shape encoder extracts rest-pose shape tokens from 2048 points randomly sampled from the object's surface and near-surface regions.

In the implementation of *Hierarchical Skeletal Modeling* (HSM) of our network, we utilize a series of set abstraction (SA) components to process and extract features from the selected 8-th frame point

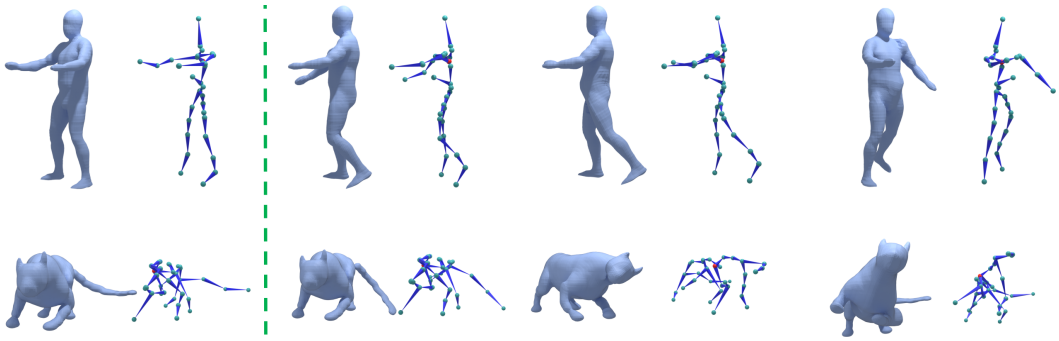

Figure S3: Editing on the D-FAUST Bogo et al. (2017) and DT4D Li et al. (2021) datasets. The left shows the rest poses with corresponding skeletons and the right illustrates the edited poses.

cloud data. For the first module, we specify radius ranges of $[0.1, 0.2]$ and sample numbers of $[16, 32]$. The multilayer perceptron (MLP) list is configured with two branches, each following the structure $16 \rightarrow 16 \rightarrow 32$. The subsequent modules gradually increase the radius ranges and sample numbers and use deeper MLPs. These modules are used sequentially for hierarchical feature extraction, and then the joint weight MLP will decode weights from these features.

# C Experiment Details

**Dataset Details:** The DFAUST Bogo et al. (2017) dataset we used for human bodies contains 10 subjects and 129 sequences. The DT4D Li et al. (2021) dataset for animals contains 38 indentities with a total of 1227 animations.
**Training Details:** We train our framework using the ADAM optimizer with a learning rate $\gamma = 10^{-4}$ and batch size of 18, which occupies about 22 GB of GPU memory with 16 compute workers. The learning rate has a step decay with a decay rate of 0.3. The training time of each epoch is 80 minutes for human bodies and 100 minutes for animals. We train 100 epochs for both.

## C.1 Data Augmentation

Our motion synthesis pipeline begins with fitting a Gaussian Mixture Model (GMM) to model the rotation Euler angles $\rho$ of reconstructed joints based on the original training dataset. The rotation angle distribution of the original training data is shown in Fig. S2. The model complexity ($M = 2$ components) was determined through silhouette analysis to ensure optimal cluster separation. Following parameter estimation, we sample 10000 rotations and visualize their distribution in Fig. S2. The rotations $\rho$ for data augmentation strategy are sampled from such a distribution and randomly applied to all reconstructed joints.

## C.2 More Results

**More Results for Motion Editing.** The inside regularization of joints and the locality constraint of skinning weight play a pivotal role in guaranteeing accurate post-editing motion. Our proposed HSM module significantly contributes to achieving an optimal joint distribution, thereby greatly facilitating subsequent build of skeleton tree and motion editing tasks. When generating a novel pose, it merely requires rotating a single joint within the skeletal hierarchy. Subsequently, the rotation of the edited joint is propagated to its corresponding child nodes via the forward kinematics technique, as schematically illustrated in Fig. S3.

**More Results for 4D Reconstruction.** Fig. S4 shows the qualitative comparison on D-FAUST Bogo et al. (2017) human bodies. Fig. S5 shows the comparison on DeformingThings4D-Animals Li et al. (2021) dataset. More animations are shown in the **Video Demo**.

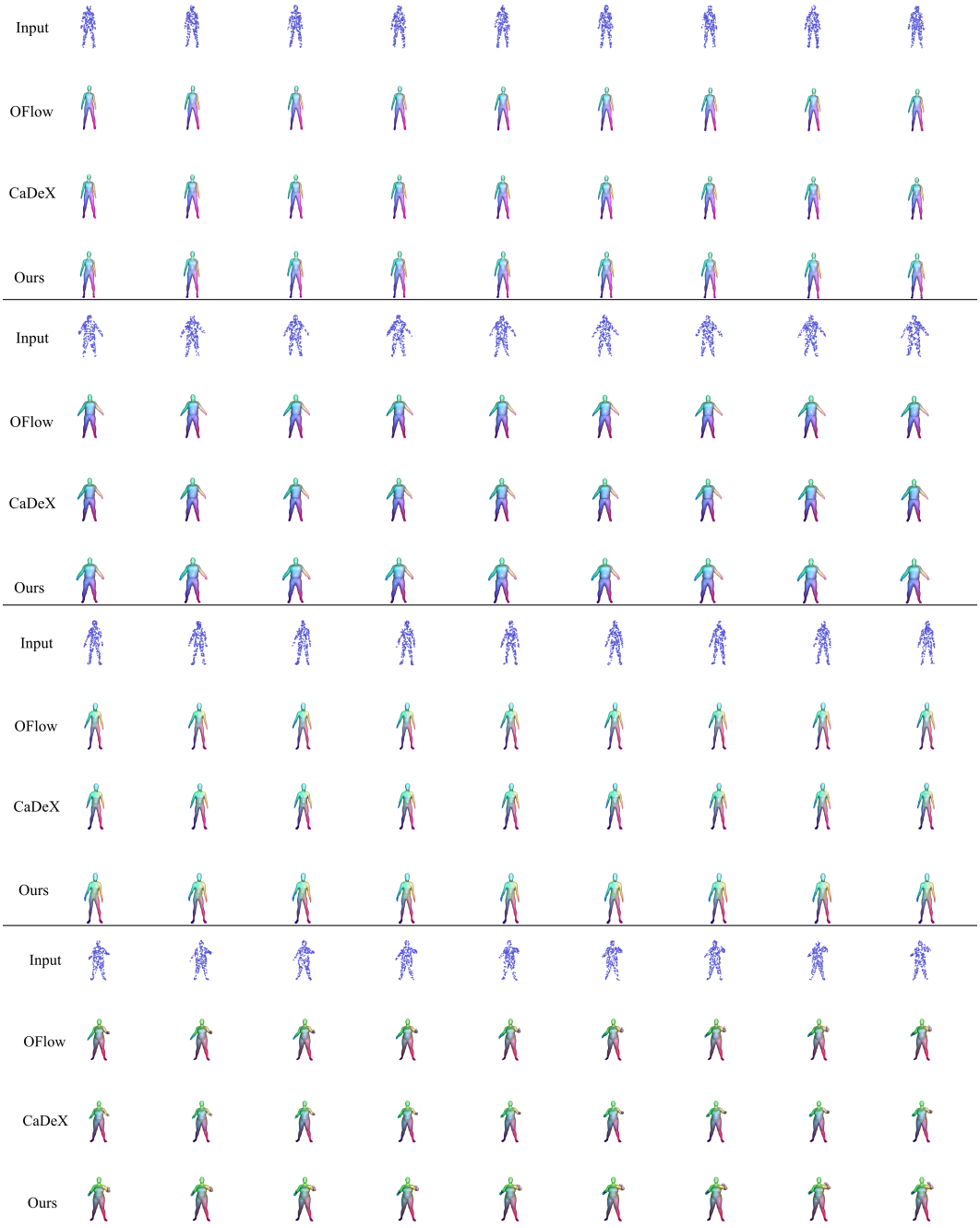


Figure S4: Modeling human bodies: each column corresponds to a time frame of the reconstruction.

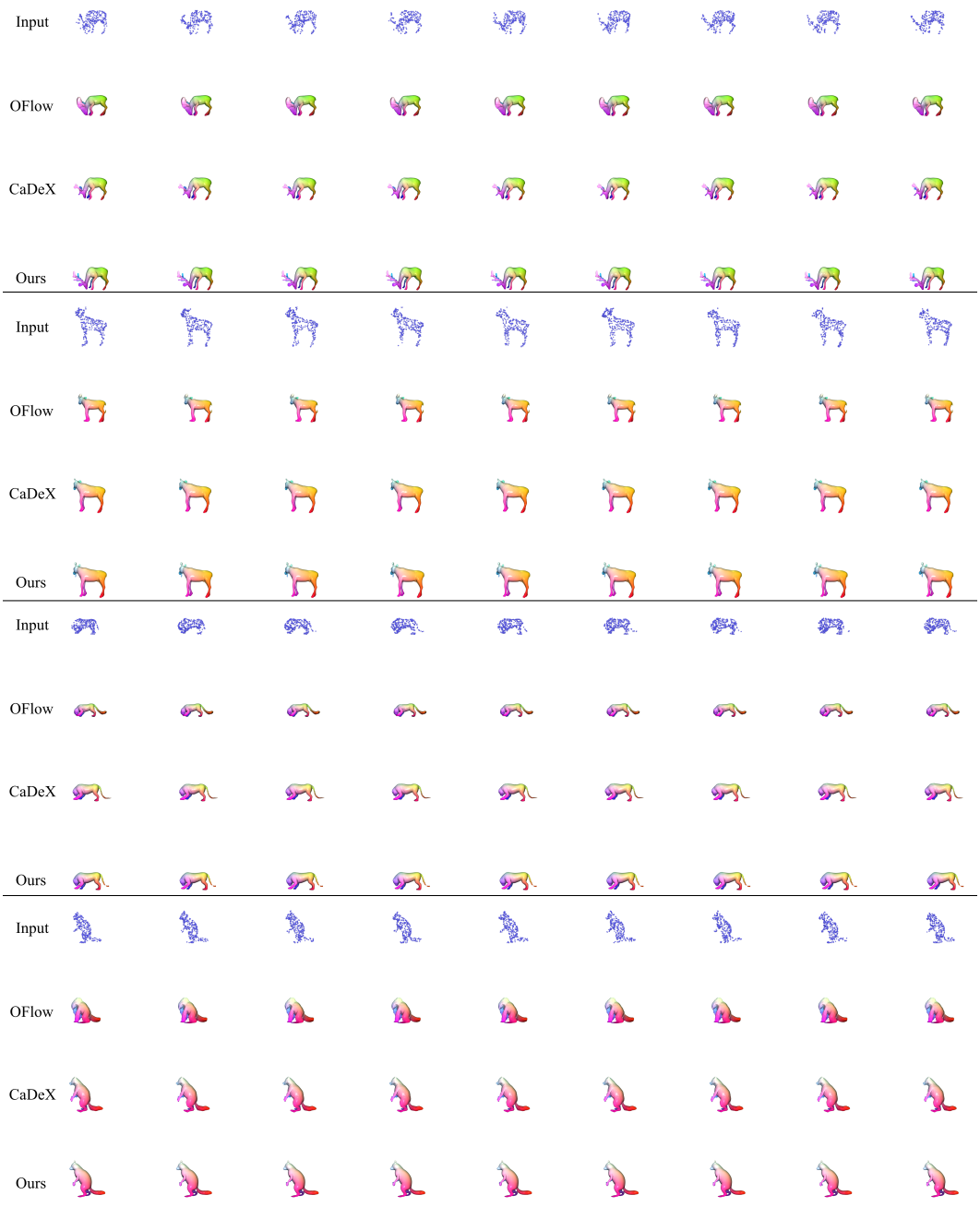


Figure S5: Modeling animals: each column corresponds to a time frame of the reconstruction.

# D Societal Benefits

Our 4D reconstruction framework offers clear societal benefits: it can accelerate applications in medical visualization (e.g. reconstructing patient-specific anatomies for surgical planning) and digital heritage preservation (efficiently digitizing artifacts). However, there are potential downsides: the same techniques could be repurposed to reconstruct people's shapes from incomplete sensor data, raising privacy and surveillance concerns.

# References


Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J Black. Dynamic faust: Registering human bodies in motion. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6233–6242, 2017.

Wei Cao, Chang Luo, Biao Zhang, Matthias Nießner, and Jiapeng Tang. Motion2vecsets: 4d latent vector set diffusion for non-rigid shape reconstruction and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20496–20506, 2024.

Jiahui Lei and Kostas Daniilidis. Cadex: Learning canonical deformation coordinate space for dynamic surface representation via neural homeomorphism. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6624–6634, 2022.

Yang Li, Hikari Takehara, Takafumi Taketomi, Bo Zheng, and Matthias Nießner. 4dcomplete: Non-rigid motion estimation beyond the observable surface. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12706–12716, 2021.

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.