AniDynRecon: Animatable 4D Dynamics Reconstruction from Sparse Point Clouds

Anonymous Author(s)
Affiliation
Address
email

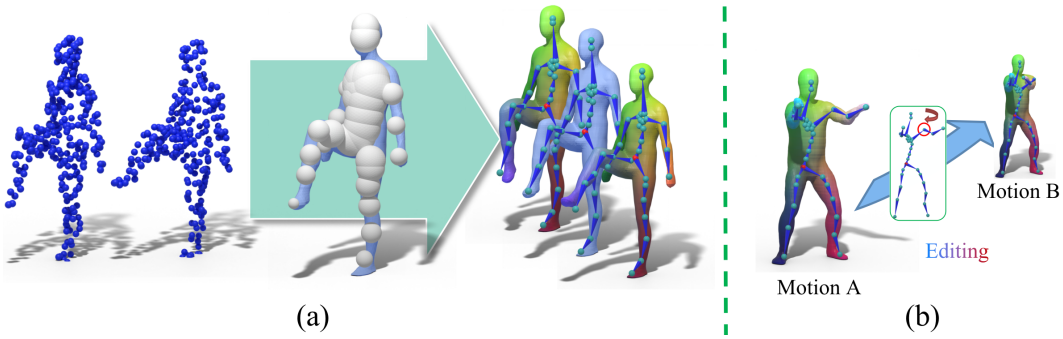


Figure 1: AniDynRecon holistically reconstructs fully animatable skeletal structures—including hierarchical topology, skinning weights, and surface geometry—directly from sparse and dynamic point clouds. (a) Reconstructed skeletal and geometric representations. (b) Motion editing results based on reconstructed skeleton structure.

Abstract

Reconstructing dynamic objects from sparse point cloud sequences without correspondence is a challenging task. Existing methods usually rely on object templates as strong geometric priors to ensure the accuracy and temporal consistency of geometric reconstruction. In this paper, we explore this highly ill-posed problem of simultaneously reconstructing object skeletons, skinning weights, motion fields, and surfaces from sparse point cloud sequences in an unsupervised manner. Specifically, we propose a *Hierarchical Skeletal Modeling (HSM)* module, which adopts a compact, interpretable skeletal abstraction inspired by the Medial Axis Transform. The skeleton representation facilitates jointly learn plausible joints, skinning weights in an unsupervised manner for capturing dynamics while reconstructing shapes. In addition, we introduce a *Combinatorial Motion Sampling (CMS)* strategy to adaptively generate new motion sequence for data augmentation utilizing the learned hierarchical skeletons. This scheme enhances the motion diversity of objects and improves the generalization of the model, especially for unseen out-of-domain scenarios. Extensive experiments on multiple benchmarks demonstrate that AniDynRecon outperforms existing methods in terms of accuracy, robustness, and structural interpretability, while enabling flexible downstream applications such as motion synthesis and editing. Our project page is https://anidynrecon-anonymous.github.io/.

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

1 Introduction

Dynamic object reconstruction is a fascinating task in computer vision and graphics, given its widespread applications across animation, AR, VR, robotics, and more. However, accurately capturing and modeling the dynamic objects from sparse point cloud data remains a significant challenge. First, it demands simultaneous modeling geometry and motion with strong temporal consistency, which is a challenging ill-posed problem. Despite significant efforts for shape modeling Loper et al. (2023); Li et al. (2017); Osman et al. (2020); Romero et al. (2022); Zuffi et al. (2017), these methods typically rely on pre-defined templates or strong assumptions about the object's structure. Neural implicit functions Palafox et al. (2021); Ma et al. (2020); Saito et al. (2021) further enhanced the power of their expression. However, reconstruction based on template categories significantly limits their generalizability and adaptability. Additionally, previous approaches Yao et al. (2025b) are frequently suffering from issues such as inaccurate joint localization, suboptimal skinning weight locality, and reduced robustness in the presence of noisy or incomplete data; See Sec. 4.1. Second, while recent advancements in deep learning have shown encouraging results Lei and Daniilidis (2022); Niemeyer et al. (2019); Cao et al. (2024); Tang et al. (2021b), these data-driven approaches are inherently limited in their generalization capability by the scope of the training data. In particular, the scarcity of sparse point cloud data further exacerbates the lack of generalization and robustness in such methods; See Sec. 5.3.

In this paper, we present AniDynRecon, a novel unsupervised end-to-end framework that reconstructs animatable skeletal structures—capturing skeletal hierarchy, skinning, and geometry—directly from dynamic point clouds. We introduce *Hierarchical Skeletal Modeling (HSM)* for dynamics reconstruction from the sparse point cloud inputs. Unlike prior work Yao et al. (2025b) that overlooks skeletal representations, we adopt a template-free skeletal abstraction inspired by the Medial Axis Transform Blum (1967) as a compact and expressive low-dimensional prior. This design offers two main advantages: (1) it effectively captures geometric structures of 3D shapes for shape recovery, as also evidenced by Lin et al. (2021); Dou et al. (2022); Tang et al. (2019) for static shape modeling, which helps obtain plausible joints without template, and (2) it can be naturally integrated with our unsupervised skinning and geometry learning modules. It enables animatable reconstructions for flexible downstream applications such as motion editing, as shown in Fig. 1. This helps generate ample motion data for training a robust model. While recent work Yao et al. (2025b) models 4D dynamics using skeletons but lacks correct joint positions for editing. In contrast, our method follows a standard animation pipeline, learning controllable joints to support motion sampling.

To overcome the challenge caused by limited 4D dynamic data, we propose a *Combinatorial Motion-Based Sampling (CMS)* strategy that augments temporal dynamics through intra-dataset motion recombination. CMS synthesizes diverse training samples by modeling the distribution of local motion segments and recombining them across different temporal windows, leveraging inherent motion variations. We show the effectiveness of CMS in significantly improving the robustness and accuracy of the learned model, especially for unseen scenarios.

We conduct extensive experiments to demonstrate that AniDynRecon outperforms state-of-the-art (SOTA) methods in accuracy, robustness, and interpretability. In summary, our contributions are as follows:

- We propose AniDynRecon, an end-to-end framework that holistically reconstructs fully animatable skeletal structures—including hierarchy, skinning, and geometry—directly from dynamic point clouds.
- We introduce *Hierarchical Skeletal Modeling (HSM)*, a compact and expressive representation inspired by the Medial Axis Transform, which unsupervisedly captures tree-structured joint hierarchies and local transformations.
- We develop a *Combinatorial Motion-Based Sampling (CMS)* strategy that leverages intra-dataset temporal variations to augment training data, thereby improving robustness under sparse or limited supervision.
- We demonstrate that AniDynRecon achieves the SOTA performance on multiple dynamic point cloud benchmarks in terms of accuracy, robustness, and structural interpretability.

2 Related Work

3D Representation 3D representations can be broadly categorized into explicit and implicit types. Explicit representations directly encode geometry, such as meshes Groueix et al. (2018); Liao et al. (2018); Pan et al. (2019); Tang et al. (2019, 2021a), point clouds Achlioptas et al. (2018); Fan et al. (2017); Zhao et al. (2021), octrees Riegler et al. (2017); Häne et al. (2017); Tatarchenko et al. (2017); Wang et al. (2018) and voxels Choy et al. (2016); Gadelha et al. (2017); Han et al. (2017); Riegler et al. (2017); Stutz and Geiger (2018). Parametric models, such as SMPL Loper et al. (2023), STAR Osman et al. (2020), FLAME Li et al. (2017), MANO Romero et al. (2022) and SMAL Zuffi et al. (2017), have been widely used to represent specific shape categories (e.g., human bodies, faces, hands and animals) through explicit geometric representations. These models leverage fixed templates and parameters to efficiently adjust pose and shape. Recently, implicit representations model 3D shapes using continuous functions, such as in Signed Distance Field (SDF) Chabra et al. (2020); Chen et al. (2023); Chen and Zhang (2019); Park et al. (2019) or occupancy networks Chibane et al. (2020); Mescheder et al. (2019). Implicit methods are more efficient and robust in handling complex shapes, as they provide smooth, continuous surfaces and can adapt to varying levels of detail. These representations are also memory-efficient and excel in tasks like shape interpolation and 3D reconstruction, particularly with noisy or sparse data. Recent advancements, such as DeepSDF Park et al. (2019) and GridPull Chen et al. (2023), demonstrate the advantages of neural implicit models in recovering high-quality 3D shapes from incomplete data, outperforming explicit methods in terms of flexibility and robustness.

Dynamic Reconstruction Recent advancements have extended traditional 3D representations into the fourth dimension (4D), enabling more effective modeling of dynamic object behaviors over time Bozic et al. (2021); Fan et al. (2017); Jiang et al. (2021); Lei and Daniilidis (2022); Niemeyer et al. (2019); Palafox et al. (2021); Tang et al. (2021b, 2022); Zou et al. (2023). Prominent works include OFLOW Niemeyer et al. (2019), LPDC Tang et al. (2021b) and CaDeX capture both shape and deformation of high quality. Recently, Motion2Vecsets Cao et al. (2024) employs a set of latent codes to represent the deformation of points. However, it is highly dependent on the correspondence between points across frames. Drawing inspiration from DynoSurf Yao et al. (2025b), which represents the deformation field using control points, our method refines the distribution of their control points and highly expands the generalization ability.

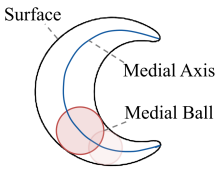
Automatic Skinning and Rigging Linear blend skinning (LBS) Le and Deng (2012) has been extensively employed for real-time animation of articulated characters. However, the manual adjustment of joints, skeleton structures, and blending weights by artists proves to be a time-consuming endeavor Bang and Lee (2018). In this regard, traditional method *Pinocchio* Baran and Popović (2007) introduced a template-based approach that enables automatic fitting of a template skeleton to a target mesh, thereby generating an animation-ready rig. Other geometric-based skinning methods exploits the theory of heat diffusion Wareham and Lasenby (2008), bounded biharmonic energy Jacobson et al. (2011), physics-inspired approaches Kavan and Sorkine (2012) and geodesic voxel binding Dionne and de Lasa (2013). Data-driven methods can capture information implied in ground-truth rigs edited by experienced animators, such as RigNet Xu et al. (2020). However, it suffers from poor robustness, which is not end-to-end trainable. HumanRig Chu et al. (2024) and Make-it-Animatable Guo et al. (2024) rely on template skeletons, which also lack the ability to rig unseen data categories.

All these approaches are unsuitable for the task of 4D reconstruction, which relies solely on point cloud, and is challenging to be trained for strong generalization. Uniquely, when coupled with the theory of MAT we are able to train a robust model for reconstructing animation-ready meshes from unseen point cloud sequences, generating diverse training samples.

3 Preliminaries

3.1 Medial Axis Transform

Medial axis transform (MAT) is a well-known method representing a closed, oriented, and bounded two-manifold surface $\mathcal{S}$ as media-axis $\mathcal{M}$ and its radius function $\mathcal{R}$ Blum (1967). The media-axis $\mathcal{M}$ is defined as the set of points in the interior with more than one closest point on the boundary surface, where the closest distance is defined as the corresponding radii $\mathcal{R}$. Each medial axis



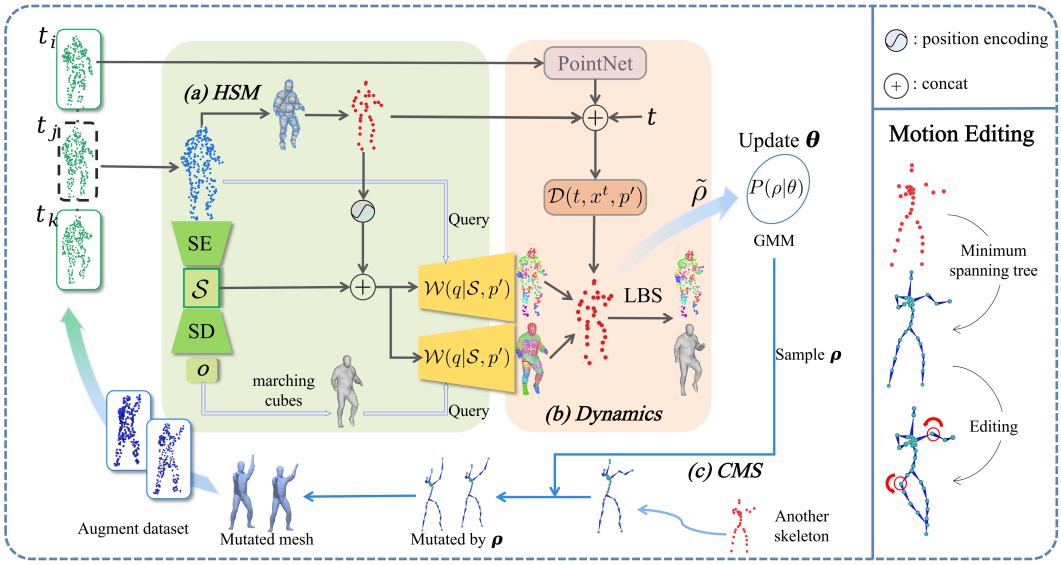


Figure 2: Overview of our method AniDynRecon. **(a) Hierarchical Skeletal Modeling (HSM)** (Sec. 4.1). Given a point cloud sequence, we initially select one frame to serve as the rest pose and encode it as shape tokens $\mathcal{S}$. Subsequently, we learn skeletal points along with their medial balls in an unsupervised manner. The occupancy field O of the rest pose is decoded from the shape tokens, and the skinning weight field can be decoded by combining the shape token with the control points. The final mesh of the rest pose is extracted from the occupancy field using marching cubes. **(b) Dynamics.** (Sec. 4.2). With skinning weights and the deformed skeletal points output from MLP $\mathcal{D}$, the mesh is deformed via LBS. The deformation results is trained to match the point cloud sequence. **(c) Combinatorial Motion-Based Sampling (CMS)** (Sec. 4.3). With the animatable reconstruction result, we model the distribution of joint rotation $\tilde{\rho}$ by fitting an GMM, then we sample new ρs to obtain new motion sequences to drive the dynamics for building an augment dataset.

point, combined with its radius, defines a medial ball. Inspired by this technique, we use the center of medial balls as control points because they have a good quality of centrality and evenly distribution across mesh.

3.2 Linear Blend Skinning

Linear Blend Skinning (LBS) Le and Deng (2012) is a widely used technique used in computer graphics and animation to simulate the deformation of a 3D mesh model, particularly when the model is subjected to skeletal animation. Given skinning weight w_{ij}, the j-th joint's initial position p'_j of rest pose, delta coordinate Δp_j^t and rotation matrix R_j^t of joint j at frame t, the deformed position of point i can be computed by LBS: $x_i^t = \sum_{j=1}^{K} w_{ij}(R_j^t d'_{ij} + p'_j + \Delta p_j^t)$, where p'_j means position of initial joints of rest pose and $d'_{ij} = x'_i - p'_j$ is the offset from joint to surface point.

4 Method

Our pipeline consists of three parts: Sec. 4.1 introduces unsupervised joint extraction via hierarchical structural modeling; Sec. 4.2 tackles joint optimization of shape, joints, skinning, and motion; Sec. 4.3 presents a synthetic-data-driven augmentation strategy to improve generalization. The overall pipeline is shown in Fig. 2.

4.1 Hierarchical Skeletal Modeling

Initial state of joints. We adopt the skeletal point prediction framework from Point2Skeleton Lin et al. (2021), which establishes skeletal points as local centroids of surface point clusters. To ensure high-quality joint extraction, we initialize the training process using a Chamfer Distance loss between predicted joints and the input point cloud in rest pose. This geometric constraint effectively distributes control points across different anatomical regions during the initial training phase.

Following joint position stabilization, we progressively introduce radius parameters r based on the concept of medial axis transform. Each sphere, centered at a joint position, is optimized to maximize contact with the rest pose surface and the corresponding radius is optimized as large as possible through the skeletal sphere loss $\mathcal{L}_{\text{centrality}}$ in Point2Skeleton Lin et al. (2021) (see the results of joints and their radii in Fig. 3):

$$\mathcal{L}_{\text{centrality}} = \left[\sum_{x \in \{x_i\}} (\min_{p' \in \{p'_j\}} ||p' - x||_2 - r(p_x^{min}) + \sum_{p} (\min_{x} ||p - x||_2 - r(p')) \right] - \sum_{p' \in \{p'_j\}} r(p') \tag{1}$$

where p_x^{min} means the closest joint to the input point x. Notably, joint positions are computed as convex combinations of surface points, with the corresponding combination weights (hereafter termed joint weights) being normalized through softmax activation:

$$\tilde{w}_{ij} = \frac{\exp(\hat{w}_{ij})}{\sum_i \exp(\hat{w}_{ij})} \tag{2}$$

where $\tilde{w}_{ij}$ means joint weight between the j-th bone and the i-th vertex and $\hat{w}_{ij}$ is the output of joint weight MLP: $\hat{w}_{ij} = \text{MLP}(f)$ (f is an embedding of rest pose encoded by PointNet++ Qi et al. (2017b)). Same way as Point2Skeleton Lin et al. (2021), we use joint weights to reconstruct the position of the joints:

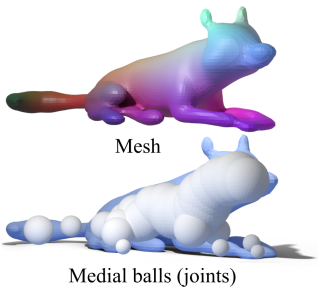


Figure 3: The medial balls approximately align with the mesh volume.

$$p'_j = \sum_{i=1}^{N} \tilde{w}_{ij} x'_i \tag{3}$$

which naturally gives the joint weights locality. (Joints tend to move towards surface points with higher weights). At the same time, we calculate the chamfer distance between the rest pose joints p' and surface points x' to ensure that the joints are evenly dispersed, which aids in the subsequent LBS. More specifically, dispersed joints prevent input point cloud regions from being covered by multiple joints with overlapping skinning weights. Another advantage of this approach is that our joints possess neutrality thanks to medial axis transform, meaning all joints are guaranteed to be distributed within the inscribed sphere of the point cloud.

Then in order to satisfy the definition of standard LBS, the joint weights should be normalized to temporary skinning weights $w'_{ij} = \tilde{w}_{ij} / \sum_j \tilde{w}_{ij}$ for further training a skinning weight field.

4.2 Jointly Train Shape, Skinning Weights and Motion

Occupancy field. We first extract a shape token $\mathcal{S} = \text{SE}(x') = \text{CrossAttn}(PE(x'), PE(x^d))$ (x^d is subsampled point cloud from x' using furthest point sampling and PE is the positional embedding) from the point cloud of the rest pose, which can subsequently be decoded into occupancy values $O = \text{SD}(q|\mathcal{S}) = \text{CrossAttn}(\text{SelfAttn}(\mathcal{S}), PE(q))$ at locations of query points q using the same shape encoder and decoder as those employed in Motion2Vecsets Cao et al. (2024). The occupancy values O, which is necessary for further mesh reconstruction using marching cubes Lorensen and Cline (1998), is supervised by ground truth values $\hat{O}$ using a BCE loss $\mathcal{L}_{\text{shape}} = \text{BCE}(O, \hat{O})$. Thanks to the occupancy field, we can use our joints as query points for shape decoder to check if they are inside the mesh volume:

$$\mathcal{L}_{\text{joints\_inside}} = \sum_{j=1}^{K} \text{BCE}(\text{SD}(p'_j|\mathcal{S}), 1) \tag{4}$$

However, the supervision of $\mathcal{L}_{\text{joints\_inside}}$ have a bad influence on the training process of shape encoder and decoder because it uses shape token $\mathcal{S}$ as condition. To avoid this, We freeze the network parameters and gradients of shape encoder and decoder before the calculation of $\mathcal{L}_{\text{joints\_inside}}$. This strategy ensures that the reconstruction of rest pose is correctly supervised and the joint positions are constrained at the same time.

Skinning weight field. To model the skinning weight field, we designed a simple skinning weight decoder, which consists of a cross-attention block and a fully connected (FC) layer. Our skinning

weight decoder takes surface points as query and joints along with shape token as key and value, outputting skinning weights w_{ij} at the query position q:

$$\mathcal{W}(q|\mathcal{S}, p') = FC(\text{CrossAttn}(PE(q), \text{Concate}(\mathcal{S}, PE(p'))) \tag{5}$$

When the query q is one point of input point cloud $q = x'_i$, the corresponding skinning weights will be $w_i = \{w_{i0}, w_{i1}, ..., w_{iK}\}$. With these skinning weights, we can supervise deformation directly on input point cloud using Eq. (7). Besides, We train our skinning weight decoder $\mathcal{W}$ based on the supervision $||w_{ij} - w'_{ij}||$ of joint weights in the early several epochs for faster convergence. Once the skinning weight field was trained and mesh of rest pose containing vertices $\bar{V}$ and triangle faces $\bar{F}$ was reconstructed from the occupancy field, we can use the vertices $\bar{V}$ to query skinning weights $\bar{W}$ on mesh surface for deformation. Given the reconstructed mesh $\{\bar{V}, \bar{F}\}$, skinning weights $\bar{W}$ correspond to its vertices and joints p at any time, it is easy to get the deformed mesh $\{\hat{V}, \bar{F}\}$ whose vertices are driven by LBS $\tilde{V} = \text{LBS}(\hat{V}, \bar{W}, p)$. Additionally, we define a locality loss to further concentrate the skinning weights around their respective joints, clearing the tiny weights corresponding to points far from the joints. The locality loss is a simple ***ReLU*** function:

$$\mathcal{L}_{\text{locality}} = \sum_{i=1}^{N} \sum_{j=1}^{K} \text{ReLU}((d'_{ij} - D)w_{ij}) \tag{6}$$

where D is a hyper-parameter denoting the maximum distance beyond which skinning weights are unconstrained. The skinning weight field is visualized in Fig. 4, 7, and an example of pose editing using our skinning weights is shown in Fig. 7.

Deformation of joints. As the deformation field of sequences is represented as deformation of joints through LBS in our framework, we use an MLP module $\mathcal{D}$ to capture dynamics contained in input point cloud sequence. Initially, we extract input point cloud features using PointNet Qi et al. (2017a) (we opt against PointNet++ Qi et al. (2017b) due to its excessive GPU memory consumption when training on point cloud sequences). Simultaneously, we apply positional embedding to encode time t and initial joint positions p'. These latent codes are then fed into a deformation MLP, yielding the joints' delta rotation and delta position $\{\Delta p^t, r^t\} = \mathcal{D}(t, x^t, p') = \text{MLP}(\text{Concate}(PE(t), PE(p'), \text{PointNet}(x^t))$ at time t. Here the rotation matrices of joints are represented as lie algebra. Finally, we utilize the chamfer distance between the point collections $\tilde{x}^t \in \{\tilde{x}_i^t\}$ of the input point cloud and output collection $x^t \in \{x_i^t\}$ of LBS in each frame as a loss function $\mathcal{L}_{\text{dynamic}}$:

$$\mathcal{L}_{\text{dynamic}} = \sum_{t=1}^{T} (\sum_{x^t \in \{x_i^t\}} \min_{\tilde{x}^t \in \{\tilde{x}_i^t\}} ||x^t - \tilde{x}^t||_2 + \sum_{\tilde{x}^t \in \{\tilde{x}_i^t\}} \min_{x^t \in \{x_i^t\}} ||\tilde{x}^t - x^t||_2) \tag{7}$$

Thanks to LBS and supervision of chamfer distance, our pipeline reconstructs dynamic mesh sequence from noisy point cloud sequence with no correspondence.

4.3 Combinatorial Motion-Based Sampling

To obtain more diverse motions for training a robust model, we sample new motions based on the original training dataset. Given joints in a novel pose, the mesh extracted from the occupancy field using marching cubes can be deformed to the new pose through LBS. AniDynRecon allows us to manually craft novel motion sequences that conform to the distribution of the training data.

However, manual editing of unstructured control points presents significant challenges. For instance, animating a simple arm movement requires individually manipulating all control points within the arm region. To address this limitation, we construct a skeleton tree using Prim's algorithm based on our joint structure, following the approach established in RigGS Yao et al. (2025a). This enables a more intuitive animation workflow: by randomly selecting a joint in the skeleton tree and applying a random rotation, child nodes are automatically transformed to new positions through forward kinematics. Instead of disturbing the synthetic data to a random distribution, we fit a Gaussian Mixture Model (GMM) for rotation euler angles ρ of joints using the original training data:

$$P(\rho|\theta) = \sum_{m=1}^{M} \pi_m \mathcal{N}(\rho|\mu_m, \Sigma_m) \tag{8}$$

Input	Method	Seen Individual				Unseen Individual		
		F-1 ↑	IoU ↑	CD ↓	Corr ↓	IoU ↑	CD ↓	Corr ↓
DT4D-A Li et al. (2021)	OFlow Niemeyer et al. (2019)	-	70.6%	0.104	0.204	57.3%	0.175	0.285
	CaDex Lei and Daniilidis (2022)	-	80.3%	0.061	**0.133**	64.7%	0.127	0.239
	DynoSurf Yao et al. (2025b)	0.401	-	0.174	0.463	-	-	-
	Ours	**0.858**	**81.7%**	**0.055**	0.166	**74.8%**	**0.073**	**0.216**
D-FAUST Bogo et al. (2017)	OFlow Niemeyer et al. (2019)	-	79.9%	0.073	0.122	69.6%	0.095	0.149
	CaDex Lei and Daniilidis (2022)	-	**85.5%**	**0.056**	**0.100**	75.4%	0.074	**0.126**
	DynoSurf Yao et al. (2025b)	0.402	-	0.118	0.175	-	-	-
	Ours	**0.854**	83.0%	0.060	0.173	**76.5%**	**0.070**	0.197

Table 1: Performance comparison on DT4D-A Li et al. (2021) and D-FAUST Bogo et al. (2017) datasets. IoU (%) indicates Intersection over Union (higher is better), CD indicates $l1$-Chamfer Distance (lower is better), and Corr indicates correspondence error (lower is better), with cell colors to indicate the **best** and second best.

where M is the number of components in the GMM, $\mathcal{N}(\rho|\mu_m, \Sigma_m)$ is the probability density function of the m-th Gaussian distribution with mean μ_m and covariance matrix Σ_m, $\theta = \{\pi, \mu, \Sigma\}$ represents the parameters for GMM. The training objective of GMM is to maximize the log-likelihood function of the observed data $\ln P(\tilde{\rho}|\theta) = \sum \ln P(\rho|\theta)$ where $\tilde{\rho} = \{\rho\}$ is the observed dataset. Once P is fitted,we sample K euler angles from GMM as the rotation of joints for synthetic pose.

5 Experiments

Datasets: Our experiments encompass two distinct categories of dynamic objects: human bodies (Sec. 5.1) and animals (Sec. 5.2). To ensure a fair comparison, we adopt the same experimental setup, datasets, and data split as CaDeX Lei and Daniilidis (2022). Firstly, human body dataset D-FAUST Bogo et al. (2017), containing 10 subjects and 109 sequences, is split into training (70%), validation (10%), and test (20%) subsets. Secondly, DeformingThings4D-Animals Li et al. (2021), including 38 indentities with a total of 1227 animations, is divided into training (75%), validation (7.5%), and test (17.5%) subsets.

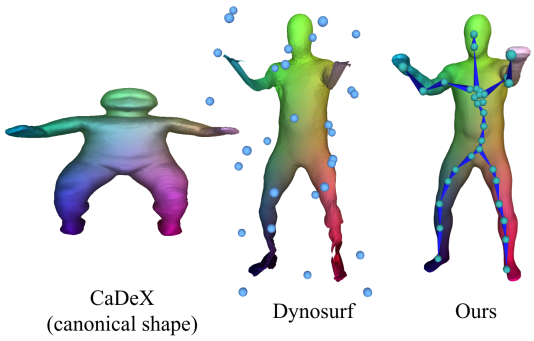


Figure 4: Visual comparison of our joints distribution with CaDeX Lei and Daniilidis (2022) and Dynosurf Yao et al. (2025b).

Baselines: We compare our method with several state-of-the-art approaches, including OFlow Niemeyer et al. (2019), CaDeX Lei and Daniilidis (2022), and DynoSurf Yao et al. (2025b). DynoSurf is the most similar to our work, as it also utilizes control points and blending weights to describe deformations.

Evaluation metrics: The Intersection over Union (IoU) metric assesses the overlap between the predicted and ground truth meshes. F-score with thresholds of 1% (F-1%). The Chamfer distance computes the average nearest-neighbor distance between two point sets. The $l2$-distance error quantifies the Euclidean distance between corresponding points on the predicted and ground truth meshes.

Runtime: The enhanced capabilities of our pipeline result in a more challenging training process compared with those aforementioned methods. To ensure the reliability and consistency of our experimental results, all the experiments are conducted on a single 4090 GPU for 10 days. We present more experiment details in the supplementary material.

5.1 Human Bodies

The input sequence of human bodies comprises 17 frames, with each frame of the input point cloud containing 300 points, evenly distributed over time to supervise deformation. The testing set encompasses two difficulty levels: unseen motion and unseen individuals Niemeyer et al. (2019). We show our results in Fig. 5. As demonstrated in Tab. 1, our method achieves advanced performance in testing on unseen individual datasets, which is more challenging than unseen motion sets. Subsequently, to

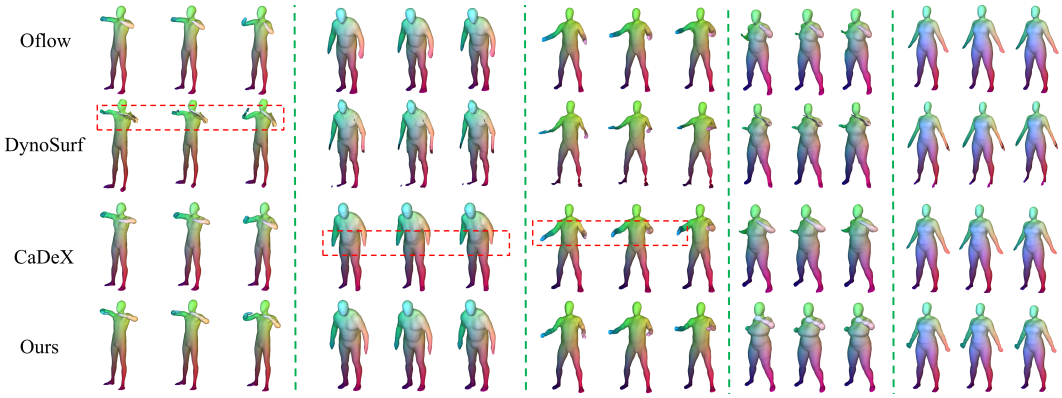


Figure 5: Comparisons on the D-FAUST Bogo et al. (2017) dataset.

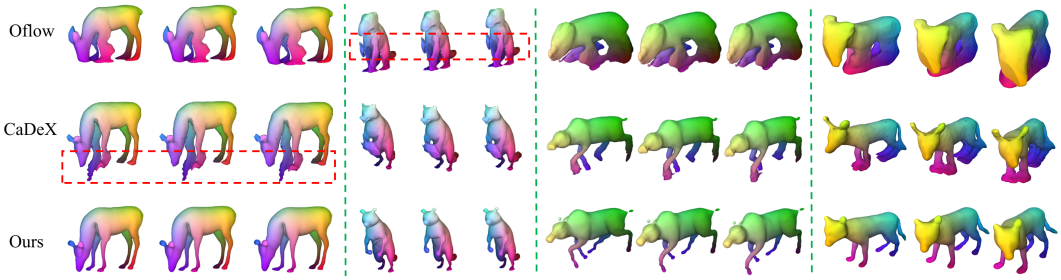


Figure 6: Comparisons on the DT4D Li et al. (2021) dataset.

make a fair comparison with DynoSurf Yao et al. (2025b) which is only trained and visualized on unseen motion datasets, we adhered to the same comparison procedure as in DynoSurf Yao et al. (2025b). The results presented in Tab. 1 highlight the advanced performance of our work against theirs.In addition to the occupancy field of shape and deformation field reconstructed by methods such as CaDeX Lei and Daniilidis (2022), our pipeline stands out by extracting control points and a skinning weight field of significantly higher quality, as shown in Fig. 4.Unlike messy canonical shapes in CaDeX Lei and Daniilidis (2022) and messy control points in DynoSurf Yao et al. (2025b), our high-quality elements can be effectively utilized in the processes of generation and motion editing.

As for the training strategy, we first train our joints with a regularization of chamfer distance loss between input point cloud and control points in the early 48 epochs. After that, the positions of joints will be optimized using the centrality loss. We refer the reader to the supplementary material for more details. It is obviously that in Fig. 4, our joints distribute evenly inside mesh but the result of DynoSurf Yao et al. (2025b) show that most of joints scatter outside the mesh irregularly.

5.2 Animals

Similar to human bodies, we utilize a 17-frame point cloud sequence as input. However, due to the greater morphological diversity across animal categories and the increased difficulty in capturing deformations compared to humans, which have a more uniform skeletal structure, we employ 512 points for deformation supervision, consistent with CaDeX Lei and Daniilidis (2022). The results is shown in Fig. 6. We present more details in the supplementary material.

Input	Method	Unseen Individual		
		IoU ↑	CD ↓	Corr ↓
D-FAUST Bogo et al. (2017)	Ours(w/o joints_inside)	76.1%	0.072	0.162
	Ours(w/o boosting)	76.5%	0.071	0.209
	Full	76.5%	0.070	0.197

Table 2: Quantitative ablation studies on the D-FAUST Bogo et al. (2017) dataset.

5.3 Ablation Study

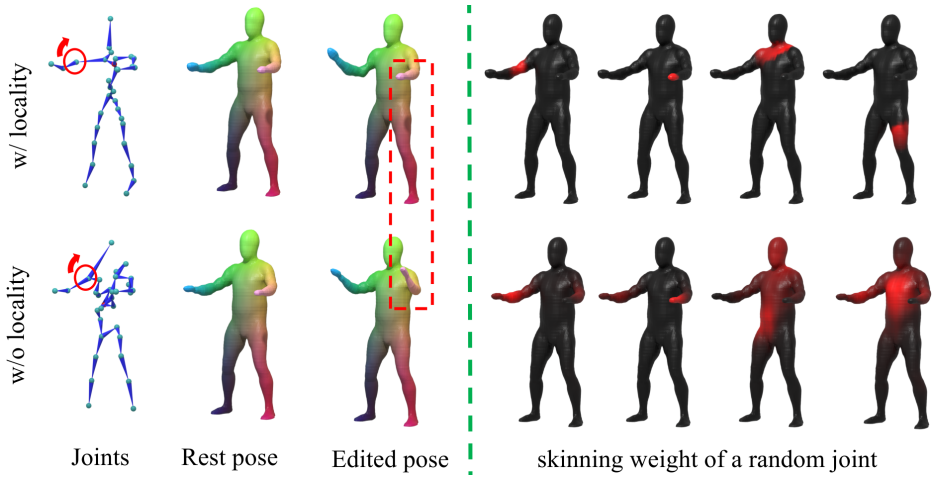


Figure 7: The locality loss $\mathcal{L}_{\text{locality}}$ ensures a good distribution of joints and skinning weights, consequently leading to correct motion editing results.

To investigate the impact of different components in our framework, we conduct ablation studies focusing on three key aspects: skinning weight locality regularization, joint position inside constraints, and data augmentation strategy.

For skinning weight locality regularization, in Fig. 7 we demonstrate that without locality loss, the distribution of joints will be messy and editing control points of one human arm will cause another arm to move, highlighting the importance of locality regularization for achieving plausible deformations. Fig. 7 also illustrates the skinning weight distribution for a random joint without locality loss supervision. For joint position inside constraints, we set the weight of joint inside loss $\mathcal{L}_{\text{joints\_inside}}$ to zero and find that some joints are generated outside the mesh, which is fatal for motion editing, as shown in Fig. 8. As indicated in Tab. 2, this negatively impacts deformation capture. Finally, we remove our data augmentation strategy and find that the correspondence metric drops, particularly for the task of unseen individuals, also shown in Tab. 2.

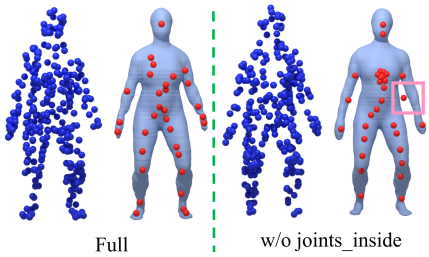


Figure 8: $\mathcal{L}_{\text{joints\_inside}}$ avoids generating joints outside of the mesh.

6 Conclusion

We present a framework named AniDynRecon for reconstructing fully animatable 4D dynamic objects directly from sparse and noisy point cloud observations. By introducing Hierarchical Skeletal Modeling (HSM), we provide an interpretable and compact representation that captures both geometric and structural information in an unsupervised manner. The integration of Combinatorial Motion-Based Sampling (CMS) further enhances the model's ability to generalize by increasing the diversity of training dynamics, especially in scenarios with limited data. Together, these components enable high-fidelity reconstruction and realistic animation, supporting downstream applications such as motion editing and synthesis. Extensive experiments demonstrate that our method consistently outperforms existing approaches in accuracy, robustness, and structural coherence. However, our method still encounters limitations in cases where real world deformations are unsuitable to be represented using LBS, such as muscle-induced elastic dynamics. This issue becomes more pronounced in animal categories, which have more uncertain biologic structures compared to human bodies. Addressing such cases is beyond the scope of this paper. Future improvements could explore how to expand the expressive capabilities of LBS using a warpping function.

References

Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pages 40–49. PMLR, 2018.

Seungbae Bang and Sung-Hee Lee. Spline interface for intuitive skinning weight editing. *ACM Transactions on Graphics (TOG)*, 37(5):1–14, 2018.

Ilya Baran and Jovan Popović. Automatic rigging and animation of 3d characters. *ACM Transactions on graphics (TOG)*, 26(3):72–es, 2007.

Harry Blum. A transformation for extracting new descriptions of shape. *Models for the perception of speech and visual form*, pages 362–380, 1967.

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.

Aljaz Bozic, Pablo Palafox, Michael Zollhofer, Justus Thies, Angela Dai, and Matthias Nießner. Neural deformation graphs for globally-consistent non-rigid reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1450–1459, 2021.

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.

Rohan Chabra, Jan E Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16*, pages 608–625. Springer, 2020.

Chao Chen, Yu-Shen Liu, and Zhizhong Han. Gridpull: Towards scalability in learning implicit representations from 3d point clouds. In *Proceedings of the ieee/cvf international conference on computer vision*, pages 18322–18334, 2023.

Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5939–5948, 2019.

Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6970–6981, 2020.

Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, pages 628–644. Springer, 2016.

Zedong Chu, Feng Xiong, Meiduo Liu, Jinzhi Zhang, Mingqi Shao, Zhaoxu Sun, Di Wang, and Mu Xu. Humanrig: Learning automatic rigging for humanoid character in a large scale dataset. *arXiv preprint arXiv:2412.02317*, 2024.

Olivier Dionne and Martin de Lasa. Geodesic voxel binding for production character meshes. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 173–180, 2013.

Zhiyang Dou, Cheng Lin, Rui Xu, Lei Yang, Shiqing Xin, Taku Komura, and Wenping Wang. Coverage axis: Inner point selection for 3d shape skeletonization. In *<Top Cited Articles in CGF 2022-2023>; EUROGRAPHICS 2022; Computer Graphics Forum 2022*, pages 419–432, 2022.

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*, pages 605–613, 2017.

Matheus Gadelha, Subhransu Maji, and Rui Wang. 3d shape induction from 2d views of multiple objects. In *2017 international conference on 3d vision (3DV)*, pages 402–411. IEEE, 2017.

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.

Zhiyang Guo, Jinxu Xiang, Kai Ma, Wengang Zhou, Houqiang Li, and Ran Zhang. Make-it-animatable: An efficient framework for authoring animation-ready 3d characters. *arXiv preprint arXiv:2411.18197*, 2024.

Xiaoguang Han, Zhen Li, Haibin Huang, Evangelos Kalogerakis, and Yizhou Yu. High-resolution shape completion using deep neural networks for global structure and local geometry inference. In *Proceedings of the IEEE international conference on computer vision*, pages 85–93, 2017.

Christian Häne, Shubham Tulsiani, and Jitendra Malik. Hierarchical surface prediction for 3d object reconstruction. In *2017 International Conference on 3D Vision (3DV)*, pages 412–420. IEEE, 2017.

Alec Jacobson, Ilya Baran, Jovan Popovic, and Olga Sorkine. Bounded biharmonic weights for real-time deformation. *ACM Trans. Graph.*, 30(4):78, 2011.

Boyan Jiang, Yinda Zhang, Xingkui Wei, Xiangyang Xue, and Yanwei Fu. Learning compositional representation for 4d captures with neural ode. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5340–5350, 2021.

Ladislav Kavan and Olga Sorkine. Elasticity-inspired deformers for character articulation. *ACM Transactions on Graphics (TOG)*, 31(6):1–8, 2012.

Binh Huy Le and Zhigang Deng. Smooth skinning decomposition with rigid bones. *ACM Transactions on Graphics (TOG)*, 31(6):1–10, 2012.

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.

Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ACM Trans. Graph.*, 36(6):194–1, 2017.

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.

Yiyi Liao, Simon Donne, and Andreas Geiger. Deep marching cubes: Learning explicit surface representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2916–2925, 2018.

Cheng Lin, Changjian Li, Yuan Liu, Nenglun Chen, Yi-King Choi, and Wenping Wang. Point2skeleton: Learning skeletal representations from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4277–4286, 2021.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 851–866. 2023.

William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pages 347–353. 1998.

Qianli Ma, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael J Black. Learning to dress 3d people in generative clothing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6469–6478, 2020.

Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019.

Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5379–5389, 2019.

Ahmed AA Osman, Timo Bolkart, and Michael J Black. Star: Sparse trained articulated human body regressor. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, pages 598–613. Springer, 2020.

Pablo Palafox, Aljaž Božič, Justus Thies, Matthias Nießner, and Angela Dai. Npms: Neural parametric models for 3d deformable shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12695–12705, 2021.

Junyi Pan, Xiaoguang Han, Weikai Chen, Jiapeng Tang, and Kui Jia. Deep mesh reconstruction from single rgb images via topology modification networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9964–9973, 2019.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019.

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, 2017a.

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, 2017b.

Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3577–3586, 2017.

Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *arXiv preprint arXiv:2201.02610*, 2022.

Shunsuke Saito, Jinlong Yang, Qianli Ma, and Michael J Black. Scanimate: Weakly supervised learning of skinned clothed avatar networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2886–2897, 2021.

David Stutz and Andreas Geiger. Learning 3d shape completion from laser scan data with weak supervision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1955–1964, 2018.

Jiapeng Tang, Xiaoguang Han, Junyi Pan, Kui Jia, and Xin Tong. A skeleton-bridged deep learning approach for generating meshes of complex topologies from single rgb images. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 4541–4550, 2019.

Jiapeng Tang, Xiaoguang Han, Mingkui Tan, Xin Tong, and Kui Jia. Skeletonnet: A topology-preserving solution for learning mesh reconstruction of object surfaces from rgb images. *IEEE transactions on pattern analysis and machine intelligence*, 44(10):6454–6471, 2021a.

Jiapeng Tang, Dan Xu, Kui Jia, and Lei Zhang. Learning parallel dense correspondence from spatio-temporal descriptors for efficient and robust 4d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6022–6031, 2021b.

Jiapeng Tang, Lev Markhasin, Bi Wang, Justus Thies, and Matthias Nießner. Neural shape deformation priors. *Advances in Neural Information Processing Systems*, 35:17117–17132, 2022.

Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *Proceedings of the IEEE international conference on computer vision*, pages 2088–2096, 2017.

Peng-Shuai Wang, Chun-Yu Sun, Yang Liu, and Xin Tong. Adaptive o-cnn: A patch-based deep representation of 3d shapes. *ACM Transactions on Graphics (TOG)*, 37(6):1–11, 2018.

Rich Wareham and Joan Lasenby. Bone glow: An improved method for the assignment of weights for mesh deformation. In *Articulated Motion and Deformable Objects: 5th International Conference, AMDO 2008, Port d'Andratx, Mallorca, Spain, July 9-11, 2008. Proceedings 5*, pages 63–71. Springer, 2008.

Zhan Xu, Yang Zhou, Evangelos Kalogerakis, Chris Landreth, and Karan Singh. Rignet: Neural rigging for articulated characters. *arXiv preprint arXiv:2005.00559*, 2020.

Yuxin Yao, Zhi Deng, and Junhui Hou. Riggs: Rigging of 3d gaussians for modeling articulated objects in videos. *arXiv preprint arXiv:2503.16822*, 2025a.

Yuxin Yao, Siyu Ren, Junhui Hou, Zhi Deng, Juyong Zhang, and Wenping Wang. Dynosurf: Neural deformation-based temporally consistent dynamic surface reconstruction. In *European Conference on Computer Vision*, pages 271–288. Springer, 2025b.

Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021.

Kaifeng Zou, Sylvain Faisan, Boyang Yu, Sébastien Valette, and Hyewon Seo. 4d facial expression diffusion model. *ACM Transactions on Multimedia Computing, Communications and Applications*, 2023.

Silvia Zuffi, Angjoo Kanazawa, David W Jacobs, and Michael J Black. 3d menagerie: Modeling the 3d shape and pose of animals. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6365–6373, 2017.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly outline the contributions, including the AniDynRecon framework, HSM, CMS strategy, and superior performance on benchmarks. These claims are substantiated in Sec. 4 (Method), 5 (Experiments), and Fig. 2.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper explicitly acknowledges limitations in its Conclusion (Sec. 6), noting that the spanning tree algorithm can produce incorrect skeleton structures in some cases and that addressing these failures is left to future work.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper focuses on methodological contributions (AniDynRecon framework, HSM, CMS) and empirical validation. It does not introduce theoretical results requiring formal proofs, but instead describes algorithmic components, loss functions, and experimental evaluations.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper details datasets (D-FAUST, DT4D-A), data splits, baselines, evaluation metrics (IoU, CD, F-score), and implementation specifics (PointNet/PointNet++ usage, GPU setup). Key components like loss functions (Sec. 4), CMS strategy (Sec. 4.3), and training phases (Sec. 5.1, 5.2) are sufficiently described. However, hyperparameters (e.g., learning rates) and full optimization details are partially covered, relying on future code release for full reproducibility.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Our code will be made publicly available upon acceptance.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper gives full experimental details, including data splits and GPU configuration in Sec. 5 and Supplementary, and specifies all key training settings, learning rate, batch size, number of epochs, plus hyperparameter values and their selection procedure in Supplementary.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: All reported metrics in the tables and figures are presented as single-point values without any accompanying error bars, confidence intervals, or description of statistical significance tests.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: The supplementary material (Implementation Details) reports that training uses 16 compute workers, occupies about 22GB of GPU memory, and requires 80 minutes per epoch for human bodies and 100 minutes per epoch for animals over 100 epochs. Additionally, All runs were conducted on a single NVIDIA RTX4090 GPU for 10 days.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: This work is purely foundational, involves no human subjects or sensitive data, poses no dual-use or safety risks, and the authors have affirmed in the NeurIPS Paper Checklist that it conforms fully to the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper have discussed potential societal impacts in supplementary material.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does not introduce or release any high-risk pretrained models, generative systems, or scraped datasets.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper credits all external assets used in the experiments, including datasets like D-FAUST and DT4D-A; citations are provided in the references section, and no violations of license terms in the text or supplementary material.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: The paper does not introduce or release any new datasets, code libraries, or pretrained model weights as standalone assets; accordingly, there is no accompanying documentation provided.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The submission contains no crowdsourcing studies or other human-subject experiments. There are no participant instructions, screenshots, or compensation details because no such research was conducted.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve any experiments with human subjects or crowd-sourced participants, and thus there are no associated risks, disclosures, or IRB approvals required or mentioned in the submission.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: According to the NeurIPS 2025 LLM policy and the Paper Checklist guidelines, LLM is used solely for writing, editing, or formatting and does not affect the core methodology, declaration is not required.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.