State Space Models for Articulated 3D Mesh Generation and Reconstruction

Supplementary Material

1. Appendix: Contents

In the following Appendices, we describe the details on the implementation, network and evaluation of MambaDiff3D (Sec. 2) and Mamba-HMR (Sec. 3). Additionally, we discuss the limitations of our technique in Sec. 4.

2. Appendix: MambaDiff3D

2.1. Network architecture and ablation setting

Our diffusion model for 3D generation is inspired by U-ViT [3] and its variants [38, 41]. It takes in the noisy 3D coordinates of surface vertices $\mathbf{x}_t \in \mathbb{R}^{N\times 3}$ and predicts noise $\epsilon_\theta^x \in \mathbb{R}^{N\times 3}$ (Fig. 1). Our MambaDiff3D consists of $L+1$ layers of Mamba blocks and input/output MLP layers. Each Mamba block contains hidden layers with d channels. The input MLP layer converts $\mathbf{x}_t$ into d-dimensional embedding features and the output MLP layer converts the Mamba-processed features into ϵ_θ^x. The time embedding corresponding to timestep t_x is incorporated to every Mamba block by summation.

The Mamba blocks are categorized into the first half shallow group with $L/2$ blocks, a mid block and a second half deep group with $L/2$ blocks. Skip connections are used to connect the blocks in the first group to those in the second group. To consider the consistencies of features flowing inside Mamba blocks, we design to use the same serialization technique for the layers connected by a skip connection in the shallow and deep group. This means at most we use $L/2+1$ serialization strategies. All of our MambaDiff3D models shown in this paper use $d=256$ and $L=12$, which means they have 13 layers in total. When we use a distinct strategy at every layer, we will have 7 different serialization strategies in our model.

Table 1 shows the ablation studies on the vertex serialization strategy. "SMPL connectivity", "part-IUV" and "TPose XYZ" indicate the serialization strategies derived from the template mesh's default vertex ordering, Densepose body part IUV maps and 3D coordinates of a T-posed template mesh, respectively. "SMPL connectivity $\times$ 1" and "part-IUV $\times$1" indicate the models using a single serialization. "part-IUV $\times$2", "part-IUV $\times$4" and "part-IUV $\times$7" are the models with two, four and seven different serialization strategies derived from the Densepose part IUV maps, respectively. "SMPL $\times$ 1 + IUV $\times$ 1", "SMPL $\times$ 1 + TPose $\times$ 1" and "SMPL $\times$ 1 + TPose $\times$ 6" mixes the different types of strategies and use the template mesh's default vertex ordering at the mid block.

Table 1. Ablation on serialization

Serialization	1NNA ↓
SMPL connectivity $\times$ 1	60.0
part-IUV $\times$ 1	54.4
part-IUV $\times$ 2	53.7
part-IUV $\times$ 4	53.7
part-IUV $\times$ 7	53.0
SMPL $\times$ 1 + IUV $\times$ 1	53.5
SMPL $\times$ 1 + TPose $\times$ 1	53.1
SMPL $\times$ 1 + TPose $\times$ 6	53.5

2.2. Comparison of network components

Table 2 and Fig. 2 shows qualitative results of unconditional 3D human generation. As illustrated in Fig. 2, MeshMamba can generate 3D human mesh models in diverse body shapes and poses. As shown in Table 2, transformer and Mamba blocks perform significantly better than MLPs and GNNs which led to unsuccessful training and produced locally very noisy surface results. Also, training was not successful with Random ordering. The generation result with the single serialization strategy based on Default ordering exhibits noise around the arms and head.

Table 2. Ablation studies on network layer blocks.

Block	1NNA ↓
MLP	73.7
GNN	74.2
Transformer	53.6
Mamba	53.1

2.3. Combining surface vertices and normals

Instead of generating 3D positions at vertices or Jacobians at triangles, we perform generation of position and normal at each vertex. Then, inspired by the techniques that transfer details in the gradient domain [5, 26, 36], we combine surface normals and positions by solving a Poisson system. This allows for smoother reconstruction by removing noise in vertices, while maintaining surface details and global shape structure (Fig. 3).

Specifically, in a similar manner as in [26], the gradient at each triangle m is obtained by combining smoothed vertex positions and surface normals in the gradient domain: $\mathbf{G}_m = \mathbf{R}_m\mathbf{T}_m$, where $\mathbf{T}_m \in \mathbb{R}^{3\times 3}$ is the Jacobian of the generated vertices after smoothing and $\mathbf{R}_m \in \mathbb{R}^{3\times 3}$ is the relative rotation between the generated normals and

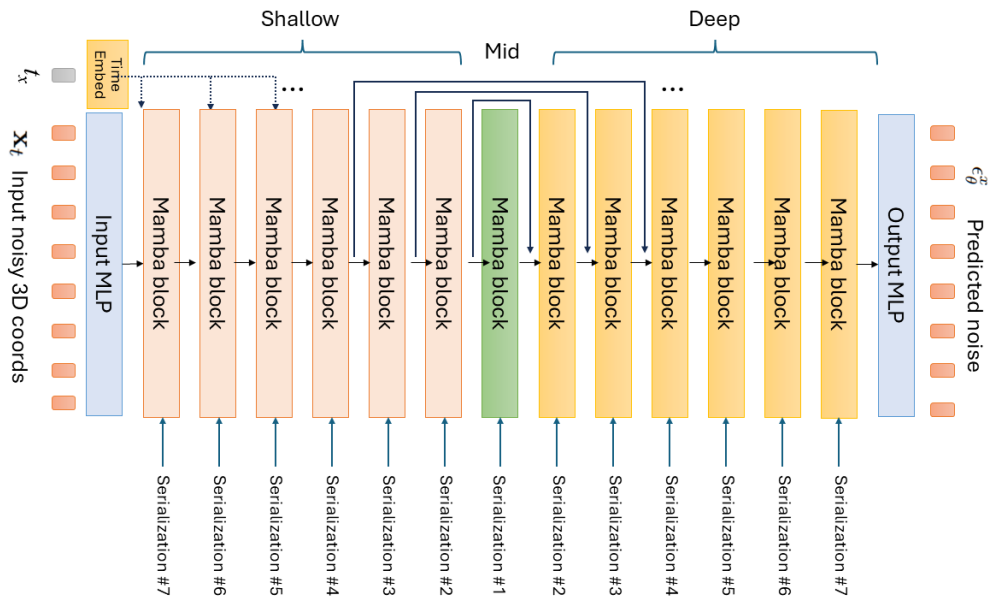

Figure 1. Network architectures of MambaDiff3D

those obtained from smoothed vertices. These gradients are then plugged in to the Poisson system [1, 32] to stich together into a whole mesh. Note that the right hand side of the Poisson system does not change for the mesh with the same connectivity. Thus, we can reuse the factorization of the system, thereby maintaining the overall generation time without a large overhead [1, 32]. Differently from previous approaches [1], our approach is not end-to-end i.e, the generation and the surface reconstruction by solving the Poisson system are done independently where no gradient is flowing from the Poisson system to the MambaDiff3D model during training.

2.4. Learning from a limited mount of data

Figure 4 compares the mesh generation results of MambaDiff3D and the transformer-based method when training on a smaller amount of data (using 4.6K training meshes). The transformer-based model generates a human mesh with arm shrinkage, resulting in large area distortions around wrists. In contrast, MeshMamba better preserves the local and global structure of the shape. We also colorized the maximum area distortion at each triangle (Fig 4 right) and plotted the percentages of triangles sorted by area distortions. As shown, MeshMamba produces fewer triangles with large distortions. This suggests that MeshMamba can be effectively trained under less-data settings by incorporating inductive biases encoded in the mesh serialization. Notably, serializing the vertices using the same vertex serialization for the transformer results in worse outcomes with significant distortions, possibly because altering the order of vertices differently for each layer makes the training of self-attention more difficult.

2.5. Additional qualitative comparison of generation results

We visualized additional generation results of MambaDiff3D and NRDF [9] in Figs. 10 and 11. We selected the first batches with 10 generated samples. We found that NRDF produces diverse but sometimes unrealistic results for unconditional generation. MambaDiff3D results are more balanced than previous studies and it can produce diverse yet realistic poses.

In Fig. 5, we also tested the serialization techniques based on z-ordering in a 3D grid as in PCM [40] and random ordering commonly used in point cloud generation [23]. These approaches produced noisy distorted results, as they neglect pose changes; for example, raising hands would lead to different serializations, disrupting mamba processing and losing correspondences.

2.6. Appendix: Additional qualitative results

Figures 12-15 visualizes additional examples of generation and reconstruction results produced by MambaDiff3D. We also provided shape interpolation results using MambaDiff3D in Figs. 16 and 17.

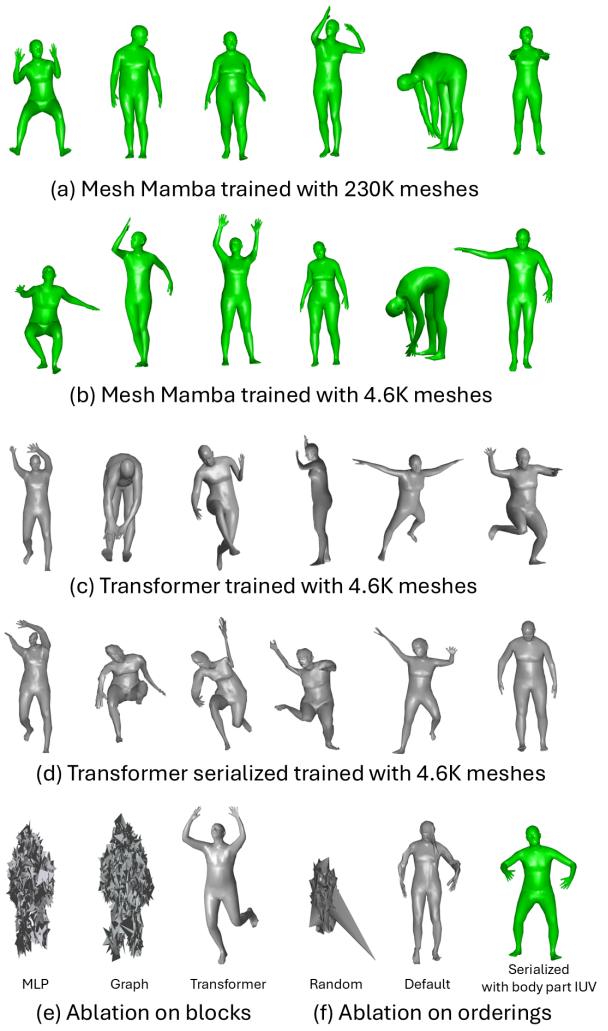

Figure 2. Qualitative comparisons network components on unconditional 3D mesh generation.

3. Appendix: Mamba-HMR

3.1. Network architectures

Network model The network architecture of Mamba-HMR follows Mesh transformer [19] where we essentially replace their transformer blocks with Mamba blocks. Our Mamba-HMR feeds CNN image features to Mamba as joint queries and vertex queries, along with position embedding. The key difference from previous vertex-based approaches [7, 15, 16, 21] is that Mamba-HMR does not necessarily need upsamplers and its Mamba-blocks directly output a full-resolution mesh, which leads to a large reduction in model parameters. The camera model and the way calculating vertex positions are same as METRO [19]. Like our MambaDiff3D, Mamba-HMR consists of the shallow, mid and deep Mamba block groups and uses skip connections, except that we do not input time embeddings. The weights in the Mamba blocks are randomly initialized.

We employ HRNet-w48 [33] and rtmpose-l [11] as our CNN backbone, initialized with the weights pre-trained on the 2D human pose detection tasks. "HR48" in Table 3 and 4 indicates the HRNet-w48 backbone model which inputs a 256×192 resolution input image and extracts an 8×6 feature map, which is pre-trained on the COCO dataset. "HR48wb" in Table 4 indicates the HRNet-w48 backbone model that uses a 384×288 image as input and extracts an 12×9 feature map, which is pre-trained on the COCO-whole body dataset [13]. rtmpose-l is pre-trained on 14 2D human pose detection dataset [12].

3.2. Further details and comparison results

We provide here more comparison results against the approaches those that are not trained on diverse datasets.

Body-only reconstruction We trained our human mesh recovery model using publicly available datasets, adopting the mixed dataset training strategies as outlined in [19, 39]. The datasets used in this paper include Human3.6M [10], MPI-INF-3DHP [24], COCO [20], MPII [2] and LSPET [14]. For training, we utilized the 3D joint labels from Human 3.6M and 2D keypoint labels from all the datasets. We use 3DPW [35] for fine-tuning our model on 3DPW following [7, 18, 19]. We conduct evaluation on Human3.6M and 3DPW.

We used the following three standard metrics for evaluation: MPJPE, PA-MPJPE and MPVE. Mean-Per-Joint-Position-Error (MPJPE) measures the Euclidean distances between the ground truth and the predicted joints. The PA-MPJPE metric, where PA stands for Procrustes Analysis, measures the reconstruction error after removing the effects of scale and rotation. Mean-Per-Vertex-Error (MPVE) measures the Euclidean distances between the ground truth and the predicted vertices.

Mamba-HMR is compared against seminal vertex-based transformer approaches: METRO [19], Graphormer [18] and FastMETRO [7]. In addition, we compared our method against more recent advanced approaches such as PointHMR [15] and HMDiff [21]. These methods regress a coarse mesh with 431 vertices using a transformer and upsample them to 6890 vertices using MLPs(sparse matrices [7] or linear layers [19]).

In Table 3, we compare our Mamba-HMR with the transformer-based baseline techniques on 3DPW and Human 3.6M. Mamba-HMR, with the smallest number of model parameters, outperforms seminal works in this field [7, 18, 19]. Notably Mamba-HMR equipped with rtmpose-l performs better than FastMetro with ResNet50 by approximately 3pts on most of the metrics and 8pts on MVE, while running at a similar inference time. Our MeshMamba is also competitive with advanced approaches such

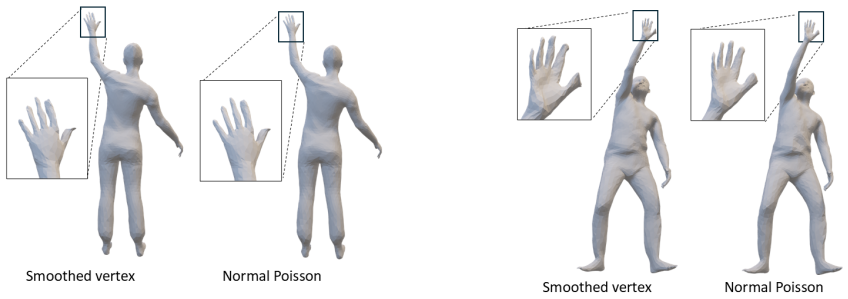

Figure 3. Comparison of our method for combining surface vertices and normals in gradient domain with a simple mesh smoothing.

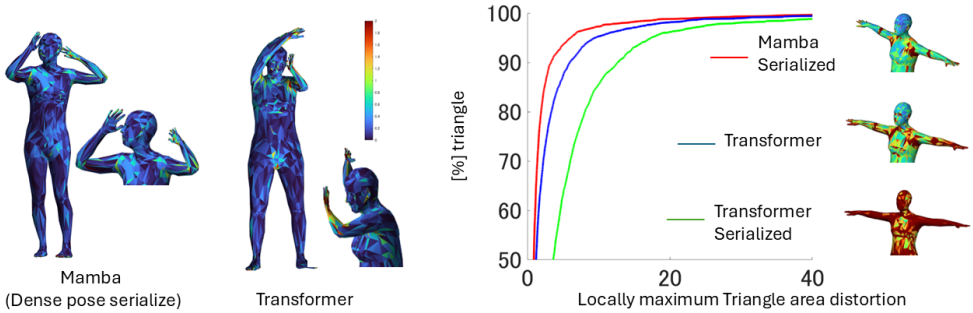

Figure 4. Comparison of Mamba and transformer under a less-data setting. Left: Visualizations of triangle area distortions for example poses generated by the two methods. The transformer-based model generates a human mesh with arm shrinkage, resulting in a large area distortions around wrists. MeshMamba better preserves the local and global structure of the shape. Right: The maximum area distortion at each triangle is colorized. The graph plots percentages of triangles sorted by area distortions. Mamba produces fewer triangles with large distortions.

as PointHMR and HMdiff. In fact, Mamba-HMR with the HRNet-w48 backbone is better than these approaches by 1-2pts on 3DPW. Furthermore, Mamba-HMR demonstrates better zero-shot transfer capability to 3DPW than METRO, likely due to its use of knowledge about local and global structure, which is extracted from a template body mesh and then encoded into mesh serialization.

Whole-body reconstruction In the first setting, Mamba-HMR is compared against previous whole-body mesh recovery approaches trained on a small number of whole-body dataset such as Pixie [8], Frankmocap [30], hand4whole [25] and OSX [17]. In this experiment, we train Mamba-HMR and OSX [17] on Human3.6M [10] and COCO [20], for the consistency of training setting. We compare against these approaches on EHF [28] and AGORA [27] without fine-tuning on them—we used EHF and AGORA in evaluation, not in training. EHF is an in-door dataset consisting of 100 images with corresponding SMPL-X labels. AGORA is a synthetic image dataset that includes severe occlusion and truncations. Our approach utilizes its validation set with 1K validation images with around 8K instances for evaluation, referenced as AGORA-val.

Tables 4 present the comparisons of whole-body mesh recovery methods on EHF and AGORA-val. On EHF and AGORA-val under no-fine tuning settings, our method is comparable to previous techniques. Although our method is inferior to the approaches which use separate image encoders for hands and face on the hand metrics, Mamba-HMR is better on the "All" metrics, indicating that its reconstruction results is high-quality overall.

In the second setting, we compare Mamba-HMR with SMPLer-X which is trained on diverse dataset. Here we train Mamba-HMR on COCO, Human3.6M, Bedlam (60K), Agora and UBody following [4, 6, 34].

As shown on Table 5, our AGORA-val PA-MVE and MVE scores are comparable to SMPLer-X-L. It is slightly worse than the SMPLer-X model trained on 32 dataset, whereas we use five dataset. For EHF, it is comparable to SMPLER-X-L5 that uses five dataset in training, but still a gap exists when they use more dataset, indicating some generalization issues still remain when tested on new dataset

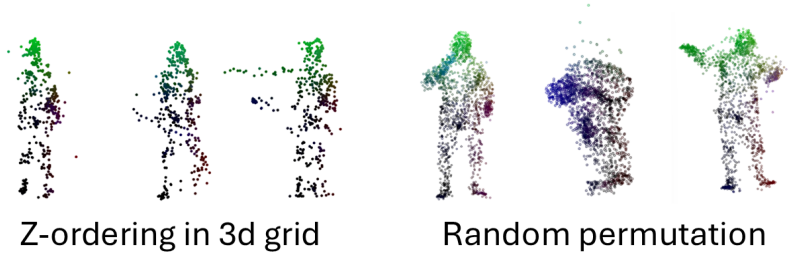

Figure 5. Generation results based on the serialization approaches used in point cloud diffusion models.

Table 3. Comparisons with vertex transformer approaches which inputs image features on 3DPW and Human 3.6M.

Method	3DPW (fine-tune)			3DPW	Human 3.6M		FPS	#Vert tokens	#Layers	#Param
	MVE ↓	MPJPE ↓	PA-MPJPE ↓	PA-MPJPE ↓	MPJPE ↓	PA-MPJPE ↓				
METRO(Res50) [19]	—	—	—	—	56.5	40.6	31.5	431	12	102.3M
METRO(HR64) [19]	88.2	77.1	47.9	63.0	54.0	36.7	13.7	431	12	230M
Mesh Graphormer [18]	87.7	74.7	45.6	—	51.2	34.5	13.6	431	12	226M
FastMETRO(Res50) [7]	90.6	77.9	48.3	—	53.9	37.3	35.3	431	12	48.4M
FastMETRO(HR64) [7]	84.1	73.5	44.6	—	52.2	33.7	14.3	431	12	153M
PointHMR [15]	85.5	73.9	44.9	—	<u>48.3</u>	<u>32.9</u>	—	431	—	—
HMDiff [21]	<u>82.4</u>	<u>72.7</u>	<u>44.5</u>	—	49.3	**32.4**	18	431	—	—
Ours (HR48)	83.6	73.2	44.7	52.8	49.6	35.9	30.3	1723	13	71.1M
Ours (HR48) w/o reg loss	—	—	—	—	**46.9**	33.5	30.3	1723	13	71.1M
Ours (HR48)	**80.5**	**71.7**	**43.8**	53.2	49.7	36.2	27.2	6890	9	70.6M
Ours (rtmpose-l)	82.8	74.9	45.3	55.1	50.9	34.4	**37.8**	6890	9	48.7M

that is not used in training especially on extreme shapes and poses. We expect that this gap could be closed by using more training dataset in future.

3.3. Details on UBody training setting

As shown in Table 7, Mamba-HMR is evaluated on UBody [17] comparing against the transformer-based parametric approaches fine-tuned on UBody or trained on various dataset, such as OSX [17], SMPLer-X [6] AiOS [34]. UBody is a dataset containing diverse real-life scenarios from movies, TV shows, talk shows, sign language, etc. Following the repository of UBody, we downsample the training set and test set to approximately 71K and 2.4K instances, respectively. The numbers of OSX [17], SMPLer-X [6] and AiOS [34] in Table 7 are taken from [34]. In the first setting, we train Mamba-HMR using Human3.6M [10] and COCO [20] with SMPL-X annotations from [25]. Then, the model is fine-tuned on UBody. In the second setting, Mamba-HMR is again trained on COCO, Human3.6M, Bedlam (60K), Agora and UBody following [4, 6, 34].

3.4. The runtime FPSs of Mamba-HMR

The runtime FPSs of Mamba-HMR are compared against other human mesh recovery approaches for body-only and whole-body settings as shown in Table 3 and 4. All timings of Table 3 were measured on an NVIDIA V100 GPU, except for HMDiff [21] which used an RTX GPU. While Mamba-HMR reconstructs a full-resolution SMPL mesh with 6980 vertices or SMPL-X mesh with 10475 vertices, it runs in (near) real-time around 30 FPS on both body-only and whole-body settings, while achieving comparable or superior mesh reconstruction accuracy to larger models of previous works. In contrast, as previous whole-body approaches rely on separate encoders for face and hands [8, 25, 29, 30] or utilize a heavy backbone encoder and decoders [17], they are limited to around 5-15 FPS.

3.5. Training loss terms

Training loss Our training loss follows [18, 19] but is augmented with local geometric losses such as the surface edge, Laplacian and normal losses, L_{edge}, L_{lap} and L_{normal} for regularization. These losses are vital for local shape preservation in our dense mesh reconstruction (Figure 6).

The total loss is defined as:

$$L = \lambda^{\text{V}}_{\text{3D}} L^{\text{V}} + \lambda^{\text{J}}_{\text{3D}}(L^{\text{J}}_{\text{3D}} + L^{\text{J}}_{\text{reg3D}}) + \lambda^{J}_{\text{2D}}(L^{\text{J}}_{\text{2D}} + L^{\text{J}}_{\text{reg2D}}) + \lambda_{\text{edge}} L^{\text{edge}} + \lambda_{\text{lap}} L_{\text{lap}} + \lambda_{\text{normal}} L_{\text{normal}} \quad (1)$$

where L^{V}, L^{J}_{3D}, $L^{\text{J}}_{\text{reg3D}}$, L^{J}_{2D} and $L^{\text{J}}_{\text{reg2D}}$ are the vertex, 3D joint, 3D regressed joint, 2D joint and 2D regressed joint loss, respectively. $\lambda^{\text{V}}_{\text{3D}}$, $\lambda^{\text{J}}_{\text{3D}}$, $\lambda^{\text{J}}_{\text{2D}}$, λ_{edge}, λ_{lap} and λ_{normal}

Table 4. Comparisons with whole-body 3D mesh recovery approaches on EHF and AGORA-val. We train Mamba-HMR and OSX [17] on Human3.6M [10] and COCO [20].

Method	EHF MVE			EHF PA-MVE			AGORA-val PA-MVE			FPS ↑	#Param	Body image Resolution	Hands&Face Experts
	All ↓	Hands↓	Face↓	All ↓	Hands↓	Face↓	All ↓	Hands ↓	Face ↓				
ExPose [29]	77.1	51.6	35.0	54.5	12.8	5.8	88.0	12.1	4.8	6.25	—	—	yes
FrankMocap [30]	107.6	**42.8**	—	57.5	12.6	—	90.6	11.2	4.9	6.7	—	—	yes
PIXIE [8]	89.2	**42.8**	32.7	55.0	**11.1**	**4.6**	82.7	12.8	5.4	10.0	—	—	yes
Hand4Whole [25]	79.2	43.2	25.0	53.1	12.1	5.8	73.2	**9.7**	**4.7**	—	—	256 × 192	yes
OSX-b [17]	78.1	58.3	27.9	52.8	16.2	6.1	72.4	11.2	**4.7**	16	185.7M	256 × 192	Decoder
OSX-l [17]	74.5	55.4	28.0	49.5	16.1	6.0	**70.7**	11.1	**4.7**	13.5	422.5M	256 × 192	Decoder
Ours (rtmpose-l)	76.5	53.9	**23.5**	49.5	16.7	6.3	71.0	10.7	5.5	**29.8**	49.8M	256 × 192	no
Ours (HR48)	80.2	62.1	23.9	52.5	16.6	6.9	71.9	10.8	6.2	27.7	66.9M	256 × 192	no
Ours (HR48wb)	**73.8**	57.4	25.2	**48.9**	16.9	5.7	71.1	10.5	**4.7**	21.3	72.3M	384 × 288	no

Table 5. Comparisons to SMPLer-X on whole-body 3D mesh recovery on AGORA-val.

Method	PA-MVE ↓ (mm)			MVE ↓ (mm)		
	All	Hands	Face	All	Hands	Face
SMPLer-X-L5 [6]	56.1	9.2	4.3	88.3	53.0	43.3
SMPLer-X-L20 [6]	48.6	8.9	4.0	80.7	51.0	41.3
SMPLer-X-L32 [6]	45.1	8.7	3.8	74.2	47.8	38.7
Ours	46.9	10.8	3.95	78.5	52.0	41.1

Table 6. Comparisons to SMPLer-X on whole-body 3D mesh recovery on EHF.

Method	PA-MVE ↓ (mm)			MVE ↓ (mm)		
	All	Hands	Face	All	Hands	Face
SMPLer-X-L5 [6]	53.9	14.7	5.9	89.5	57.8	29.9
SMPLer-X-L20 [6]	37.8	15.0	5.1	65.4	49.4	17.4
SMPLer-X-L32 [6]	37.1	14.1	5.0	62.4	47.1	17.0
Ours	47.3	17.8	5.8	72.9	54.4	23.5

Table 7. Comparisons with whole-body 3D mesh recovery approaches on UBody. † indicates fine-tuned on UBody. ♯ indicates trained on Human3.6M and COCO. ♭ indicates trained on Human3.6M, COCO, Bedlam, Agora and UBody.

Method	PA-MVE ↓ (mm)			MVE ↓ (mm)			
	All	Hands	Face	All	Hands	Face	FPS
OSX-L [17]	42.4	10.8	2.4	92.4	47.7	24.9	14
OSX-L [17] †	42.2	8.6	**2.0**	81.9	41.5	21.2	14
SMPLer-X-L [6]	33.2	10.6	2.8	61.5	43.3	23.1	24
SMPLer-X-L [6] †	31.9	10.3	2.8	57.4	40.2	21.6	24
AiOS [34]	32.5	**7.3**	2.8	58.6	39.0	19.6	—
Multi-HMR-B [4]	31.4	9.8	6.1	65.1	**33.1**	22.6	23
NLF-L [31]	66.8	19.4	6.6	—	—	—	**41**
Ours ♯	45.3	10.7	3.3	107.6	49.7	31.7	22
Ours ♭	26.3	10.7	2.4	54.4	38.8	17.7	22
Ours †	**25.9**	9.7	2.1	**51.7**	33.9	**15.9**	22

are the weights for controlling the relative strengths of respective terms.

3D vertex loss Let $\mathbf{v} \in \mathbb{R}^{N\times 3}$ be the predicted vertex positions. The vertex loss L^V is defined with a L1 loss function as follows:

$$L^V = \frac{1}{N}(||\mathbf{v} - \bar{\mathbf{v}}||_1) \tag{2}$$

where $\bar{\mathbf{v}} \in \mathbb{R}^{N\times 3}$ is the ground truth vertex positions.

3D joint loss Similarly, the 3D joint loss L^J_{3D} can be defined from the output joint locations in image space and model space, $\mathbf{J}_{3\mathrm{D}}$ as:

$$L^J_{3D} = \frac{1}{J}(||\mathbf{J}_{3\mathrm{D}} - \bar{\mathbf{J}}_{3\mathrm{D}}||_2) \tag{3}$$

where $\bar{\mathbf{J}}_{3\mathrm{D}} \in \mathbb{R}^{J\times 3}$ is the ground truth joint positions.

Regressed 3D joint loss The 3D joint locations can also be calculated from the output mesh using a pre-computed joint regressor matrix [19, 22]. Let $\mathbf{J}^{\mathrm{reg}}_{3\mathrm{D}} \in \mathbb{R}^{J\times 3}$ be the 3D joint locations obtained with the pre-computed regressor. Then the regressed 3D vertex loss L^J_{reg3D} is defined as follows:

$$L^J_{\mathrm{reg3D}} = \frac{1}{J}(||\mathbf{J}^{\mathrm{reg}}_{3\mathrm{D}} - \bar{\mathbf{J}}_{3\mathrm{D}}||_2) \tag{4}$$

2D joint loss 2D supervisions are useful whenever 3D supervision is not available, such as when the input is an in-the-wild image. Given the 2D joint estimates $\mathbf{J}_{2\mathrm{D}} \in \mathbb{R}^{J\times 2}$ and the 2D regressed joints $\mathbf{J}^{\mathrm{reg}}_{2\mathrm{D}} \in \mathbb{R}^{J\times 2}$, the 2D joint loss L^J_{2D} is defined by:

$$L^J_{2D} = \frac{1}{J}(||\mathbf{J}_{2\mathrm{D}} - \bar{\mathbf{J}}_{2\mathrm{D}}||_2 + ||\mathbf{J}^{\mathrm{reg}}_{2\mathrm{D}} - -\bar{\mathbf{J}}_{2\mathrm{D}}||_2 \tag{5}$$

where $\bar{\mathbf{J}}_{2\mathrm{D}} \in \mathbb{R}^{J\times 2}$ is the ground truth 2D joint locations.

The regularization terms are defined from the ground truth and predicted SMPL or SMPL-X mesh vertices.

Edge loss The edge loss L_{edge} is defined as:

$$L_{\mathrm{edge}} = \frac{1}{E}\sum_{e=1}^{E} ||\mathbf{e}_e - \bar{\mathbf{e}}_e||_1 \tag{6}$$

where $\mathbf{e}_e$ and $\bar{\mathbf{e}}_e$ are the predicted and ground truth of edge length at edge e, respectively.

Laplacian loss The Laplacian loss L_{lap} is written as:

$$L_{\mathrm{lap}} = \frac{1}{N}\sum_{i=1}^{N} ||\mathbf{d}_i - \bar{\mathbf{d}}_i||_1 \tag{7}$$

where $\mathbf{d}_i$ and $\bar{\mathbf{d}}_i$ are the predicted and ground truth of mean curvature normal vector at vertex i derived from the cotangent Laplacian matrix, respectively.

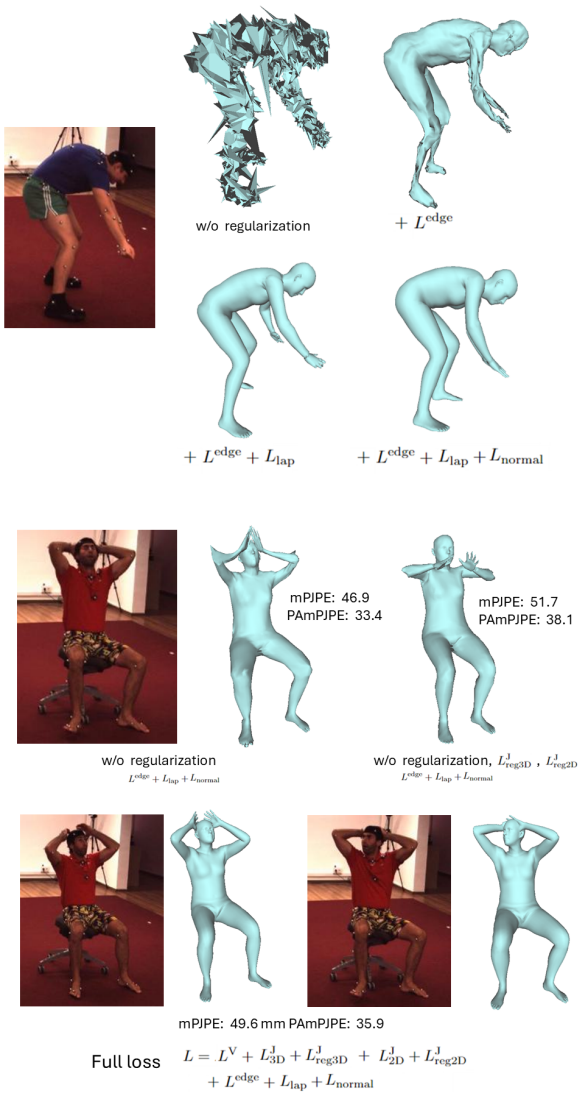

Figure 6. Ablation on training losses. Top: varying regularization terms when using 6890 vertex tokens. Bottom: varying training losses using 1723 vertex tokens.

Normal loss The normal loss L_{normal} is defined as:

$$L_{\text{normal}} = \frac{1}{F} \sum_{m=1}^{F} ||\mathbf{n}_m - \bar{\mathbf{n}}_m||_1 \tag{8}$$

where $\mathbf{n}_m$ and $\bar{\mathbf{n}}_m$ are the predicted and ground truth of face normal at triangle face m, respectively.

3.6. Ablation on the training losses

Figure 6 visualizes human mesh recovery results by varying the training losses. For reconstructing a full-resolution dense mesh with 6890 vertex tokens, the regularization terms are vital (Figure 6 top). When all the regularization terms are removed, it results in a very noisy reconstruction. Adding each regularization term improves the smoothness and shape preservation of reconstructed meshes.

For reconstructing a down-sampled mesh with 1723 vertex tokens, it is possible to reconstruct smoother meshes without enforcing the regularization terms, as with the previous vertex transformer approaches [19]. This leads to high accuracy on body joint prediction but the reconstructed mesh is prone to large distortions especially around joints (Fig. 6 bottom). By removing the loss for fitting the regressed 3D joints, the geometry of a resulting mesh is maintained locally. However, the mesh does not properly align with body joints. The incorporation of the regularization terms remedy this issue and achieves comparable MPJPE and PA-MPJPE scores.

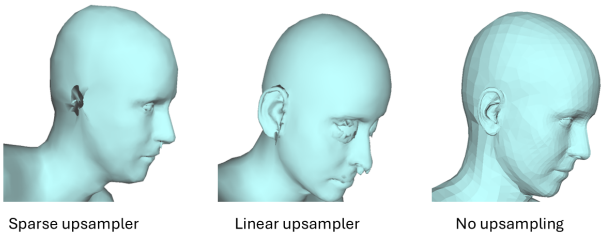

Figure 7. Comparisons against using a sparse upsampler [7] and linear upsampler [19]. Mamba-HMR does not use an upsampler and directly outputs a full resolution mesh. Methods using an upsampler faces difficulties in reconstructing ears and eyes for whole body reconstruction based on a SMPL-X mesh model due to its complex geometry and topology around the head.

3.7. Comparison of whole-body reconstruction results w/o VS. with upsampling

Figure 7 shows the comparisons against using a sparse upsampler [7] or linear upsampler [19]. Mamba-HMR does not use an upsampler and directly outputs a full resolution mesh. The methods using an upsampler faces difficulties in reconstructing ears and eyes for whole-body reconstruction based on a SMPL-X mesh model due to its complex geometry and topology around the head.

3.8. Additional qualitative results

Additional examples of reconstruction results produced by Mamba-HMR are visualized in Figs. 8 and 9.

4. Appendix: Discussions and future work

As discussed in the main text, the strength of MeshMamba is its efficiency and inference speed, while increasing the number of vertex tokens to enable the generation and reconstruction of dense 3D meshes. In Sections 2.4 and 3.7, we

also provide some analysis results on why MeshMamba can achieve superior results to previous approaches. There, we show that MeshMamba 1) can incorporate inductive biases through its vertex serialization and 2) can eliminate the need of an upsampling process that sometimes produces erroneous results. We believe that these aspects of MeshMamba are especially beneficial for the situations where the training data is scarce such as the 3D domain. Furthermore, a combination of MeshMamba with the gradient domain surface integration and training losses that incorporates surface normals is a promising strategy to obtain a smooth yet detailed surface results. By leveraging MeshMamba, MambaDiff3D outperformed other generative models in the 3D human generation task and Mamba-HMR showed its strong performance on the whole body reconstruction task under the fine-tuning setting by extending previous vertex-transformer approaches.

MeshMamba still has limitations that could be addressed in future research. First, it is currently limited to tight clothing with a fixed topology. We aim to tackle more challenging in-the-wild clothed human mesh recovery tasks [37] by further increasing image and mesh resolution. Additionally, investigating a way to incorporate hand and finger experts into Mamba-HMR in a general and efficient manner would be valuable attempts to further improve reconstruction of finger poses and facial expressions. Second, its generalization capability to new datasets not used in training is still limited compared to approaches trained on diverse datasets [6]. while MeshMamba possesses a relatively good zero-shot transfer ability compared to nonparametric transformer approaches, currently it may not match with that of parametric approaches or low-dimensional latent approaches. Future work could explore a way to train MeshMamba on a larger-scale dataset collections like in [6] and assess its generalization ability under such a larger-scale training setting.

References

[1] Noam Aigerman, Kunal Gupta, Vladimir G Kim, Siddhartha Chaudhuri, Jun Saito, and Thibault Groueix. Neural jacobian fields: Learning intrinsic mappings of arbitrary meshes. *SIGGRAPH*, 2022. 2

[2] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Schiele Bernt. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 2014. 3

[3] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *CVPR*, 2023. 1

[4] Fabien Baradel*, Matthieu Armando, Salma Galaaoui, Romain Brégier, Philippe Weinzaepfel, Grégory Rogez, and Thomas Lucas*. Multi-hmr: Multi-person whole-body human mesh recovery in a single shot. In *ECCV*, 2024. 4, 5, 6

[5] Mario Botsch, Robert W. Sumner, Mark Pauly, and Markus Gross. Deformation transfer for detail-preserving surface editing. 2006. 1

[6] Zhongang Cai, Wanqi Yin, Ailing Zeng, Chen Wei, Qingping Sun, Yanjun Wang, Hui En Pang, Haiyi Mei, Mingyuan Zhang, Lei Zhang, et al. Smpler-x: Scaling up expressive human pose and shape estimation. In *NeurIPs*, 2023. 4, 5, 6, 8

[7] Junhyeong Cho, Kim Youwang, and Tae-Hyun Oh. Cross-attention of disentangled modalities for 3d human mesh recovery with transformers. In *ECCV*, 2022. 3, 5, 7

[8] Yao Feng, Vasileios Choutas, Timo Bolkart, Dimitrios Tzionas, and Michael Black. Collaborative regression of expressive bodies using moderation. In *3DV*, pages 792–804, 2021. 4, 5, 6

[9] Yannan He, Garvita Tiwari, Tolga Birdal, Jan Eric Lenssen, and Gerard Pons-Moll. Nrdf: Neural riemannian distance fields for learning articulated pose priors. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[10] 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 TPAMI*, 36(7):1325–1339, 2014. 3, 4, 5, 6

[11] Tao Jiang, Peng Lu, Li Zhang, Ningsheng Ma, Rui Han, Chengqi Lyu, Yining Li, and Kai Chen. Rtmpose: Real-time multi-person pose estimation based on mmpose, 2023. 3

[12] Tao Jiang, Xinchen Xie, and Yining Li. Rtmw: Real-time multi-person 2d and 3d whole-body pose estimation, 2024. 3

[13] Sheng Jin, Lumin Xu, Jin Xu, Can Wang, Wentao Liu, Chen Qian, Wanli Ouyang, and Ping Luo. Whole-body human pose estimation in the wild. In *ECCV*, 2020. 3

[14] Sam Johnson and Mark Everingham. Learning effective human pose estimation from inaccurate annotation. In *CVPR*, pages 1465–1472, 2011. 3

[15] Jeonghwan Kim, Mi-Gyeong Gwon, Hyunwoo Park, Hyukmin Kwon, Gi-Mun Um, and Wonjun Kim. Sampling is Matter: Point-guided 3d human mesh reconstruction. In *CVPR*, 2023. 3, 5

[16] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *CVPR*, 2019. 3

[17] Jing Lin, Ailing Zeng, Haoqian Wang, Lei Zhang, and Yu Li. One-stage 3d whole-body mesh recovery with component aware transformer. 2023. 4, 5, 6

[18] Kevin Lin, Lijuan Wang, and Zicheng Liu. Mesh graphormer. In *ICCV*, 2021. 3, 5

[19] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *CVPR*, 2021. 3, 5, 6, 7

[20] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. 3, 4, 5, 6

[21] Hossein Rahmani Jun Liu Lin Geng Foo, Jia Gong. Distribution-aligned diffusion for human mesh recovery. In *ICCV*, 2023. 3, 5

Figure 8. 3D human mesh recovery results on Human 3.6M, 3DPW, EHF and AGORA

[22] 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, 2015. 6

[23] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[24] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *3D Vision (3DV), 2017 Fifth International Conference on*. IEEE, 2017. 3

[25] Gyeongsik Moon, Hongsuk Choi, and Kyoung Mu Lee. Accurate 3d hand pose estimation for whole-body 3d human mesh estimation. In *CVPRW*, 2022. 4, 5, 6

[26] Diego Nehab, Szymon Rusinkiewicz, James Davis, and Ravi Ramamoorthi. Efficiently combining positions and normals for precise 3d geometry. *ACM Trans. Graph.*, 24(3): 536–543, 2005. 1

[27] Priyanka Patel, Chun-Hao P. Huang, Joachim Tesch, David T. Hoffmann, Shashank Tripathi, and Michael J. Black. AGORA: Avatars in geography optimized for regression analysis. In *Proceedings IEEE/CVF Conf. on Computer*

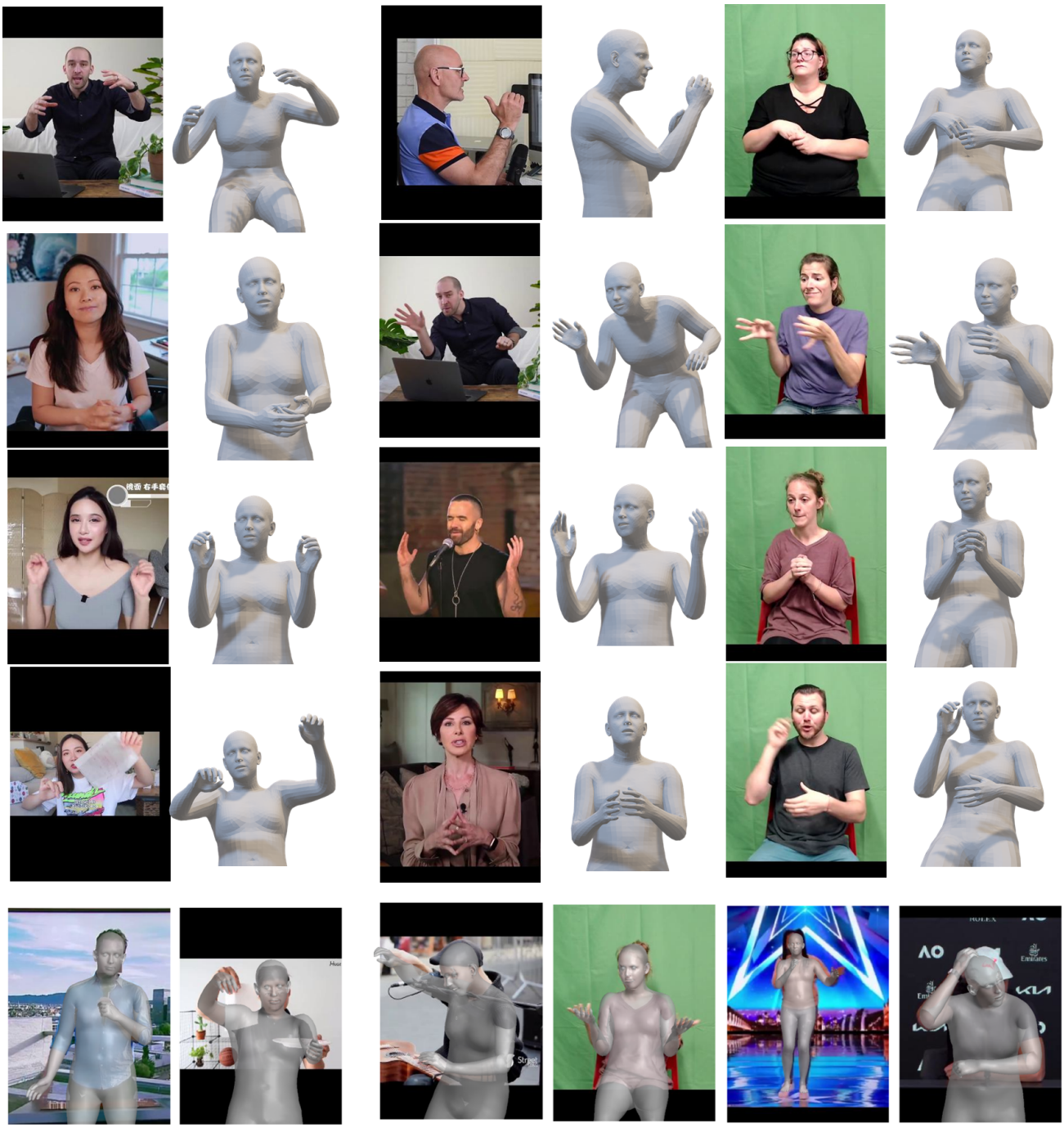

Figure 9. Whole-body 3D human mesh recovery from a single image on UBody

Vision and Pattern Recognition (CVPR), 2021. 4

[28] 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. 4

[29] Georgios Pavlakos, Vasileios Choutas, Timo Bolkart, Dimitrios Tzionas, Michael J. Black, Vasileios Choutas, Georgios Pavlakos, Timo Bolkart, Dimitrios Tzionas, and Michael J. Black. Monocular expressive body regression through body-driven attention. *ECCV*, 2020. 5, 6

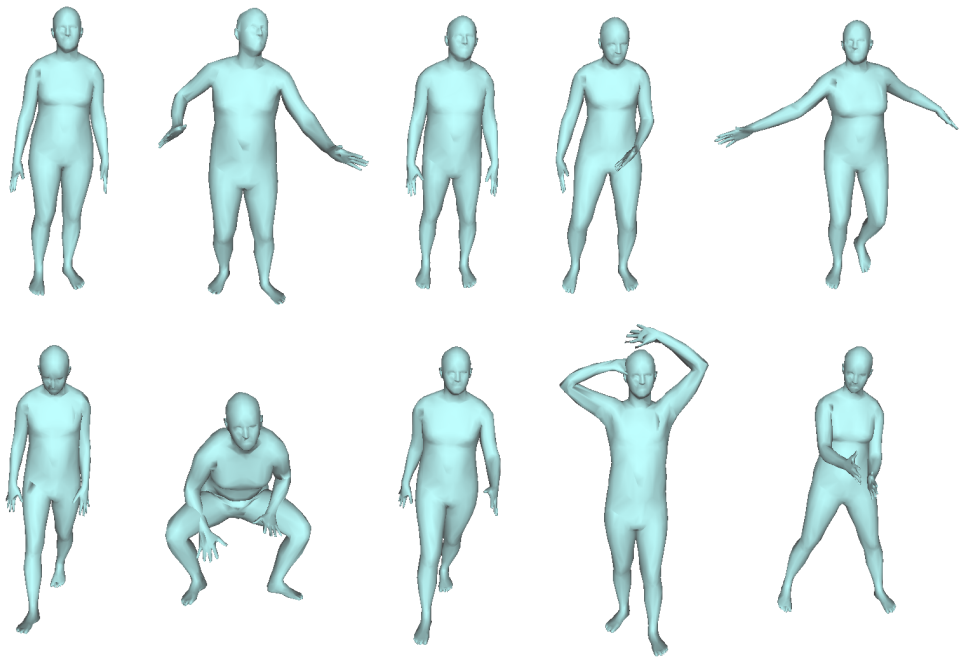

Figure 10. MambaDiff3D human mesh generation on AMASS (1723 vertex tokens)

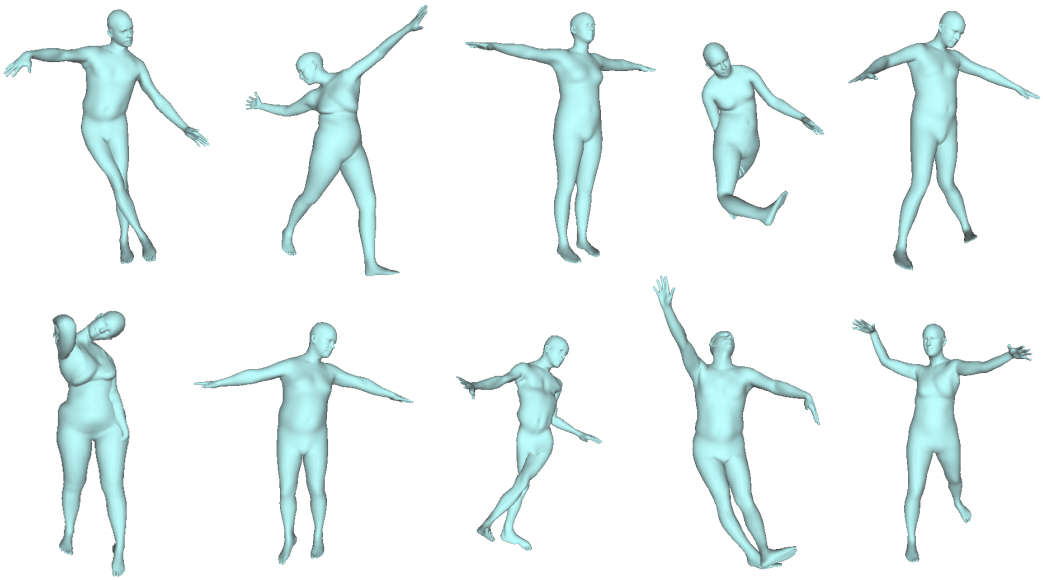

Figure 11. NRDF generation results on AMASS

[30] Yu Rong, Takaaki Shiratori, and Hanbyul Joo. Frankmocap: A monocular 3d whole-body pose estimation system via regression and integration. In *ICCVW*, 2021. 4, 5, 6

[31] István Sárándi and Gerard Pons-Moll. Neural localizer fields for continuous 3d human pose and shape estimation. 2024. 6

[32] Robert W. Sumner and Jovan Popović. Deformation transfer for triangle meshes. *SIGGRAPH*, 23(3), 2004. 2

[33] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose esti-

Figure 12. Unconditional 3D mesh generation results on CAPE

mation. In *CVPR*, 2019. 3

[34] Qingping Sun, Yanjun Wang, Ailing Zeng, Wanqi Yin, Chen Wei, Wenjia Wang, Haiyi Mei, Chi Sing Leung, Ziwei Liu, Lei Yang, and Zhongang Cai. Aios: All-in-one-stage expressive human pose and shape estimation. In *CVPR*, 2024. 4, 5, 6

[35] Timo von Marcard, Roberto Henschel, Michael Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *ECCV*, 2018. 3

[36] Ofir Weber, Olga Sorkine, Yaron Lipman, and Craig Gotsman. Context-aware skeletal shape deformation. *Computer Graphics Forum*, 26(3):265–274, 2007. 1

[37] Yuliang Xiu, Jinlong Yang, Dimitrios Tzionas, and Michael J. Black. ICON: Implicit Clothed humans Obtained from Normals. In *Proceedings of the IEEE/CVF Conference*

Figure 13. Unconditional 3D mesh generation results on CAPE

on Computer Vision and Pattern Recognition (CVPR), pages 13296–13306, 2022. 8

[38] Yusuke Yoshiyasu and Leyuan Sun. Diffsurf: A transformer-based diffusion model for generating and reconstructing 3d surfaces in pose. In *ECCV*, 2024. 1

[39] Hongwen Zhang, Yating Tian, Xinchi Zhou, Wanli Ouyang, Yebin Liu, Limin Wang, and Zhenan Sun. Pymaf: 3d human pose and shape regression with pyramidal mesh alignment feedback loop. In *ICCV*, 2021. 3

[40] Tao Zhang, Xiangtai Li, Haobo Yuan, Shunping Ji, and Shuicheng Yan. Point cloud mamba: Point cloud learning via state space model. *arXiv preprint arXiv:2403.00762*, 2024. 2

[41] Changqian Yu Jusnshi Huang Zhengcong Fei, Mingyuan Fan. Scalable diffusion models with state space backbone. *arXiv preprint*, 2024. 1

Figure 14. Unconditional 3D mesh generation results on GRAB

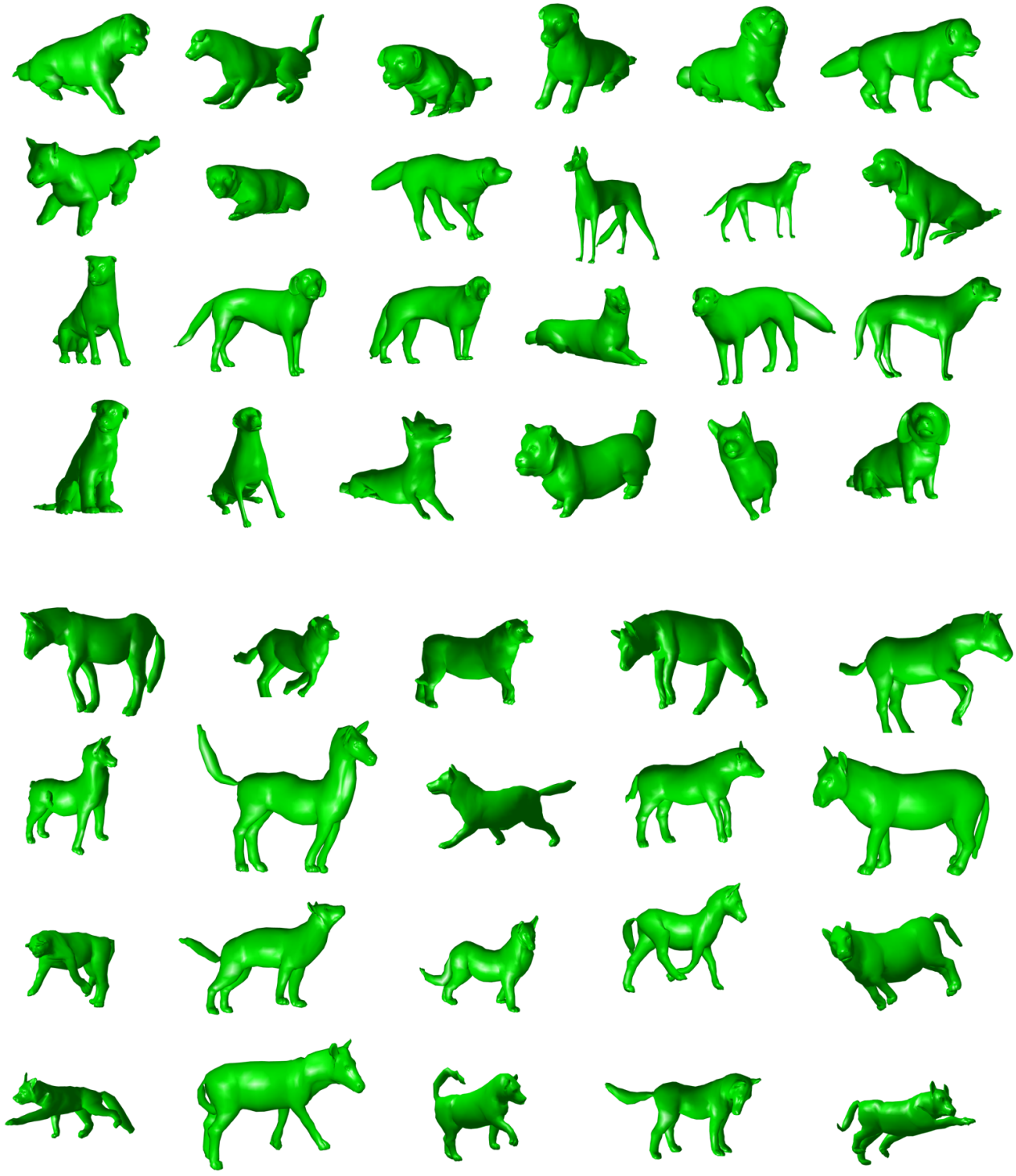

Figure 15. Unconditional 3D mesh generation results on BARC and Animal3D

Figure 16. Shape interpolation results on DFAUST

Figure 17. Shape interpolation results on Surreal