FRESA: Feedforward Reconstruction of Personalized Skinned Avatars from Few Images

Supplementary Material

A. Data Acquisition

Data Capture. We visualize our dome capture data in Figure 1. For each subject, we capture 128 images from different view points using a group of fixed cameras, and adopt SuperNormal [3] to reconstruct the 3D scans for geometry supervision. We further estimate 3D joints for each frame from multi-view images, and adopt an incremental pose encoder [4] to obtain the pose vector $\boldsymbol{p}$. With the diverse posed clothed humans and high-quality scans, we can learn an effective universal prior that well generalizes to phone photos.

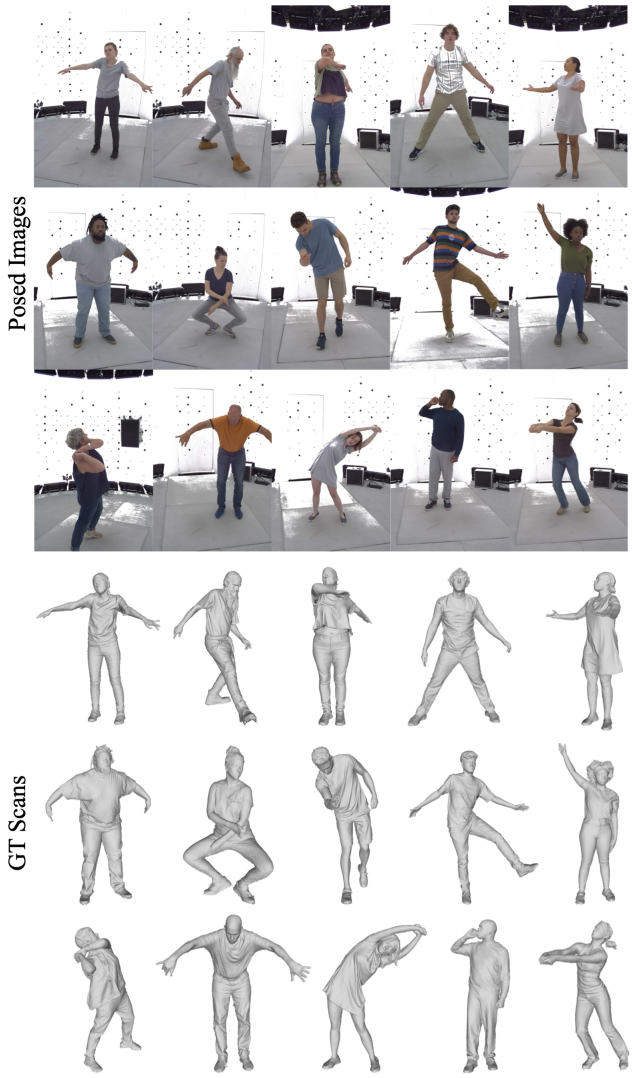

Figure 1. **Samples of dome data.** Our dataset contains diverse posed clothed humans paired with high-quality 3D scans as ground truths, which facilitates learning an effective universal prior.

Pseudo-GT Canonical Meshes. To resolve the coupled ambiguity between canonical shapes and skinning weights, we construct pseudo-GT[1] canonical meshes as the regularization for canonical-space stage training. Specifically, we adopt FlexiCubes [16] as the 3D geometry representation and build a cubic grid with $G = 256^3$ vertices near the rigged body template. We then initialize the SDF value s of each grid vertex as the signed distance to the template, and optimize canonical vertices $\bar{\mathbf{V}}$ parameterized by the Flexicube parameters $s, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}$ [16] such that:

$$\bar{\mathbf{V}} = \arg\min_{\mathbf{V}}(||\mathcal{R}_d(\tilde{\mathbf{V}}) - \hat{\mathcal{D}}^\star||_1 + ||\mathcal{R}_n(\tilde{\mathbf{V}}) - \hat{\mathcal{N}}^\star||_1 + \mathcal{L}_r) \ , \tag{1}$$

where $\mathcal{R}_d(\cdot)$ and $\mathcal{R}_n(\cdot)$ are renderer functions for normal and depth rendering, $\tilde{\mathbf{V}}$ is the posed vertices obtained by forward LBS (where skinning weights queried from nearest template vertices), $\mathcal{L}_r$ is the regularization term in [16], and $\hat{\mathcal{D}}^\star, \hat{\mathcal{N}}^\star$ are ground truth depth and normal images for posed scans. We empirically observe that such unposing strategy reduces artifacts of over-stretched triangles as shown in Figure 2. However, since such unposing requires a complete optimization process and takes 20 minutes to converge, it is only suitable for data preparation.

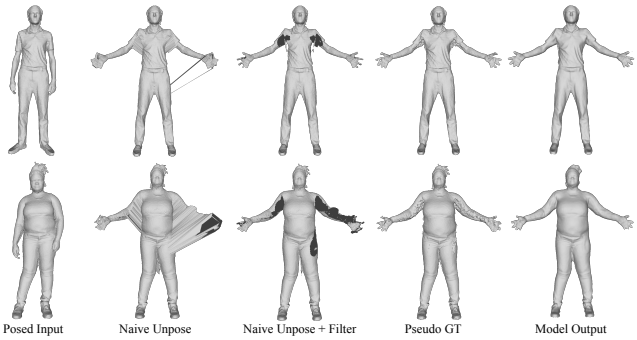

Figure 2. **Unposing Comparison.** We compare the results between naive unposing (used in the inference pipeline) and pseudo GT via optimization (used for data preparation). The second approach produces more plausible results but requires significantly more time. Note we filter edges with length larger than 1×10^{-4} to reduce noises.

Approval of Usage. All participants involved in dome capture data and phone photos have signed a consent form that authorize the usage of their images for model development and academic publications.

[1] ideal GT meshes should be obtained by unposing scans with personalized skinning weights, which is not available in canonical-space stage.

B. Canonicalization Details

Pose Tracking for Photos. We use an artists designed rigged body mesh as the template, which contains $J = 67$ joints. For each front and back view photo, we estimate its 2D joint positions using [9], and optimize the pose vector $\boldsymbol{p}$ to minimize the 2D projection loss similar to [2]. We further determine the absolute scale of the subject based on a pretrained statistical prior model using PCA coefficients. The overall optimization process takes about 1 minute per frame. For a fair comparison, in the main paper, we report inference time for all methods excluding the pose tracking time as we assume known poses in our pipeline.

Note that to ensure a practical use of our method, we do not require a perfect alignment between front and back views for causally taken photos, *i.e.* we do not require known camera poses, and photos do not need to be synchronized in time, as illustrated in Figure 3. Such casual inputs can be robustly handled with the universal clothed human prior and multi-frame aggregation, as shown in main paper. In addition, since there are no GT body poses for phone photos, we use an off-the-shelf pose estimator to estimate body poses for each view. Demo results in the main paper show that our method can robustly generate plausible avatars under this imperfect unposing, ensuring a practical use of our method. Moreover, the proposed multi-frame aggregation approach can further improve robustness against inaccurate pose estimation in individual frame.

Figure 3. **Illustration of settings for photos.** We use *estimated* body poses and do not require perfect alignment between views.

3D Lifting. We follow [19] to use d-BiNI method to obtain the lifted front and back surface meshes for each frame. The surface depth is initialized based on the tracked poses, *i.e.* the surface depth of posed body template. We visualize the resulting unposed surface meshes in Figure 4.

In contrast to [19] that attempts to directly complete the lifted meshes in the 3D space, we re-render them into 2D images as initial conditions, and infer the canonical shape *from scratch* for two reasons: (*i*) it produces a more plausible boundaries by jointly refining geometry in both visible and invisible parts, as shown in Figure 5. (*ii*) it can learn a personalized body shape instead of a fixed shape bounded by the initial depth, as shown in Figure 6. Finally, we choose to use two views in the paper as the lifting and pose tracking process work mostly robust in these two views. However, our method can still produce a plausible side view geometry by learning across diverse subjects.

Figure 4. **Illustration of Lifted Surface Meshes.** Note we removed the over-stretched edges after unposing. The lifting process produces two unposed surface meshes but can not be perfectly aligned in boundary.

C. More Implementation Details

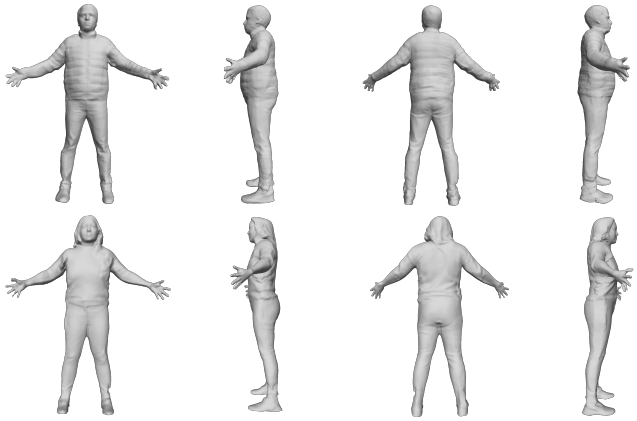

Figure 5. **Visualization in Four Views.** By only taking inputs of front and back views, our method can infer plausible side-view geometry and produce a consistent boundary.

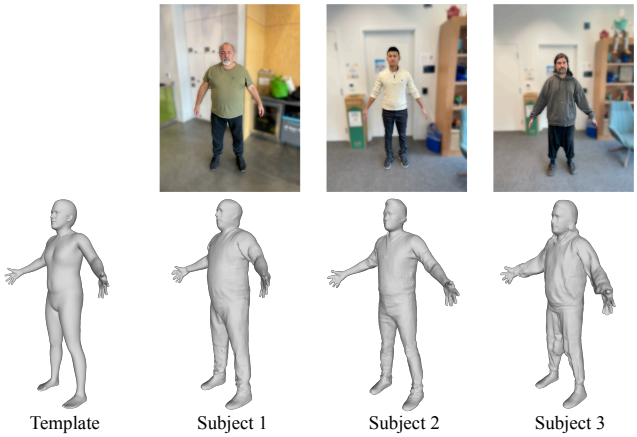

Figure 6. **Results of Inferred Body Shape.** Our method can produce personalized body shapes based on input conditions and is not restricted to the template shape.

We train our model using $N = 5$ input frames, which achieves the best balance between plausibility and fidelity. In inference, our model can be applied to an arbitrary num-

ber of input frames based on availability. To increase the model generalizability, we apply a data augmentation by mixing unposing results from both d-BiNI and 3D scans when training the model. Finally, we use [1] to tetrahedralize the volume near the canonical template (with a distance of 0.2m), resulting in a grid of resolution 256. More details about our network architecture are included in Appendix D.

Inference Time Details. We report the inference time for one input photo with image size 1280×960. Specifically, (*i*) the segmentation and normal estimation for [9] takes 4.90s, (*ii*) the d-BiNI time for both views takes 9.91s, (*iii*) the unposing (including finding the nearest template vertices) takes 1.54s, (*iv*) the canonical rendering takes 0.06s, and (*v*) the overall model inference takes 1.64s, thus the total inference time is 18.05s. All time are reported with a single NVIDIA A100 GPU.

Baseline Implementation. For [7], we use the test code and pretrained models provided by the author. Since the method only takes a single image as input, we test its reconstruction quality by using the target posed image as input. For [6, 20], we modify its code to use our rigged template and canonical pose instead of SMPL-X [12] template. For [6], we also follow their implementation to use nearest template vertices' skinning weights, weighted by the point-to-point distances in deformed space, while [20] only uses a template mesh to initialize the DMTet grid.

D. Network Architecture

Multi-Frame Encoder. We show the architecture for $f_e(\cdot)$ in Figure 11. For simplicity of notation, we refer to all input features as these after concatenating the front and back views, *e.g.* $\bar{\mathbf{N}} \in \mathbb{R}^{2\times512\times512\times3}$, and thus discarding the view dependency in the superscript and assume all features below have a batch size of 2. In $f_l(\cdot)$, the DeepLabV3 [5] backbone produce a feature map of shape $64 \times 64 \times 256$, and the output channels for the Conv2d are [128, 128, 96] respectively. All upsampling blocks are implemented as $2\times$ bilinear interpolation, thus the dimension for $\mathbf{L}_i$ is $256\times256\times96$. In $f_h(\cdot)$, the output channels for the Conv2d are [64, 96, 96, 96, 96] respectively. Note we follow [17] to include positional encoding before the first convolution block, thus its input channel is 6 instead of 8. Except that the first convolution block in $f_h(\cdot)$ has a stride 3, all other blocks have a stride 1. The final dimension for $\mathbf{H}_i$ is the same as $\mathbf{L}_i$. In $f_b(\cdot)$ the output channels for the Conv2d are [256, 128, 96], and the biplane feature $\mathbf{B}_i$ has a shape of $512 \times 512 \times 96$.

Canonical Geometry Decoder. We show the architecture for $f_g(\cdot)$ in Figure 12. For each grid vertex $\boldsymbol{g}$, we sample the feature on $\mathbf{B}_i$ to obtain the feature $\boldsymbol{\phi} \in \mathbb{R}^{96}$. The output channels for each Linear block is [64, 64, 64, 64, 4]. Note here we include BatchNorm (BN) [8] and treat each vertex as a batch sample. We observe this module can be used to replace geometric initialization [21] to ensure a valid mesh at initial steps, *i.e.* avoid situations where the network predicts all positive or negative SDF values.

Skinning Weight Decoder. We show the architecture for $f_s(\cdot)$ in Figure 12. Except that the last linear layer has an output dimension of 161, all other modules follow the same architecture as $f_g(\cdot)$.

Pose-dependent Deformation Decoder. We show the architecture for $f_c(\cdot)$ in Figure 12. We first render front and back position maps $\mathbf{P}_t \in \mathbb{R}^{512\times512\times3}$ follow [11] and concatenate with the rendered front and back images of the inferred canonical mesh, and forward it to $f_d(\cdot)$ (with the same architecture as $f_e(\cdot)$) to produce a residual biplane $\hat{\mathbf{B}}_t \in \mathbb{R}^{512\times512\times96}$. We then sample pixel feature $\boldsymbol{\psi}_t \in \mathbb{R}^{96}$ as the feature for each vertex in the canonical mesh. The output channels for each Linear block is [64, 64, 64, 64, 3].

E. More Animation Comparison

In this section, we compare with SCANimate [15], which optimizes personalized skinning weights and canonical shapes jointly in an implicit field. As shown in Figure 7, while such approach produces smooth deformation, the use of implicit field results in low geometry resolution and thus missing fine-grained details. Moreover, [15] rely on time-consuming per-subject fitting and 3D posed meshes as inputs, whereas our method can achieve *instant* feed-forward reconstruction from *few images*.

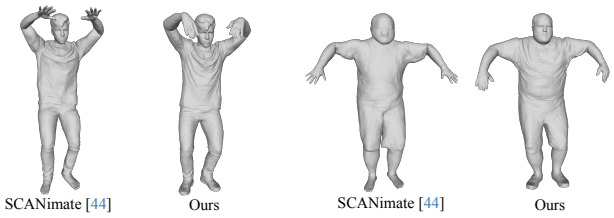

Figure 7. **Animation comparison with SCANimate.** For [15], we use FRESA reconstructions as reference posed meshes. Note that hand motions are missing as it is SMPL-based.

Comparison with Reconstruction Methods

In this work, we aim to generate personalized avatars that can be *realistically animated* driven by *novel poses*. In contrast, other baselines like [10, 13, 14, 18, 19, 22] are often characterized as single-image reconstruction method, which focuses on recovering the geometry for the *input pose* only, and animates posed avatars using a fixed skinning weights. Hence they do not study avatar animation and thus are *not closely related to our work*. Moreover, in the experiments we evaluate the animation quality on *unseen* poses and predict pose-dependent deformation to recover fine-grained details like wrinkles, thus a fair comparison is difficult to perform with the reconstruction methods. For completeness,

we show in Figure 8 that our method can produce high-quality geometry details comparable to [10, 19, 22], thanks to the effective prior learned from diverse subjects. Considering fairness of evaluation, we do not quantitatively benchmarking reconstruction quality on our dataset.

Figure 8. **Qualitative comparison with single image reconstruction methods.** Our method produces high-quality geometry details comparable to ECON [19], SIFU [22], and PSHuman [10] on both dome data and phone photos.

F. Texture Reconstruction

Our method can be extended to generate texture for the reconstructed personalized meshes. In this section we provide one sample implementation for texture reconstruction. Specifically, we first unpose lifted surface meshes with back-projected vertex color (refer to Figure 4 as an example) and render the RGB images as input. We then encode the RGB images into a separate bi-plane feature (using a encoder with the same architecture as $f_e(\cdot)$), and pose the canonical avatar by the target pose vector $\boldsymbol{p}_t$. For each rendered pixel of the posed avatar mesh, we use the corresponding 3D position on the canonical mesh to sample the bi-plane feature, which is forwarded to a MLP decoder (with the same architecture as $f_c(\cdot)$) to predict the RGB color for that pixel. We show in Figure 9 that this approach produces realistic rendering results.

Figure 9. **Results of Textured Meshes.** Our method can be extended to produce high-resolution texture for realistic rendering.

G. Failure Cases

In our method, the deformation module is only conditioned on a skeletal pose vector, which is deficient to model complex dynamics such as motions of hair or extremely loose garments like a long dress. We show failure cases in Figure 10, where the results posed deformation do not match the real dynamics. Future works are encouraged to explore more comprehensive pose conditions or physics-inspired models to tackle this issue.

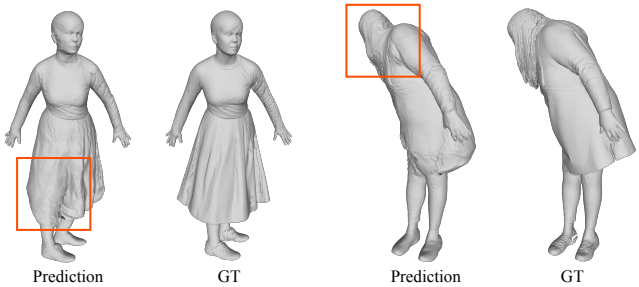

Figure 10. **Failure Cases.** With only the pose vector as condition, our method fails to produce complex hair motions and dynamics of extremely loose garments.

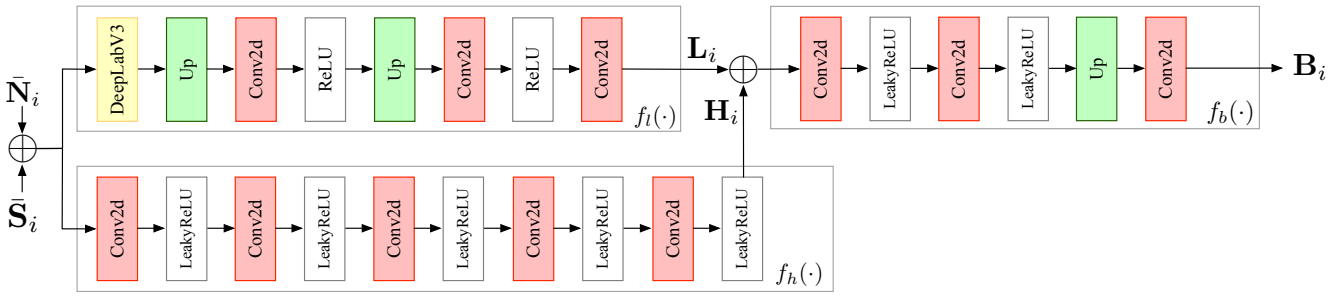

Figure 11. **Model Architecture for multi-frame encoder $f_e(\cdot)$.** Note we stack two views together and omit the superscript v. The final bi-plane feature is obtained by summing the feature for each frame $\mathbf{B}_i$. $\oplus$ denotes channel-wise concatenation.

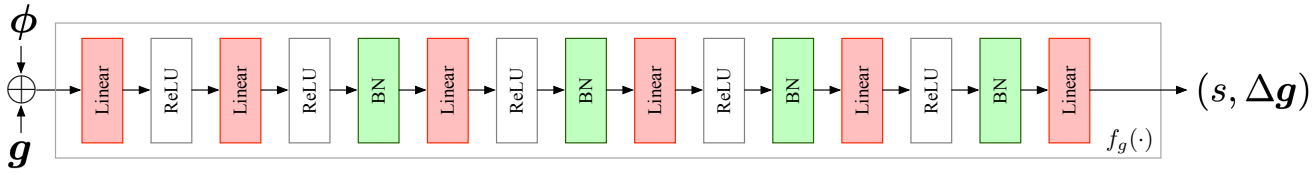

Figure 12. **Model Architecture for canonical geometry decoder $f_g(\cdot)$.**

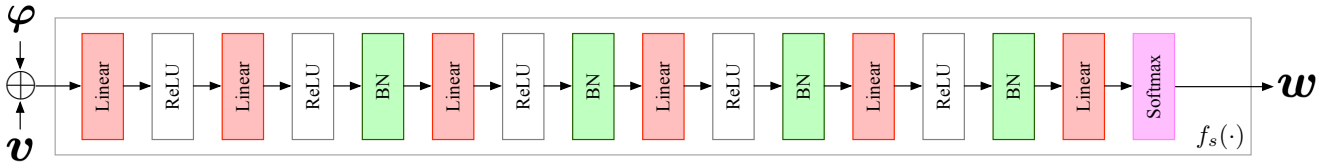

Figure 13. **Model Architecture for skinning weight decoder $f_s(\cdot)$.**

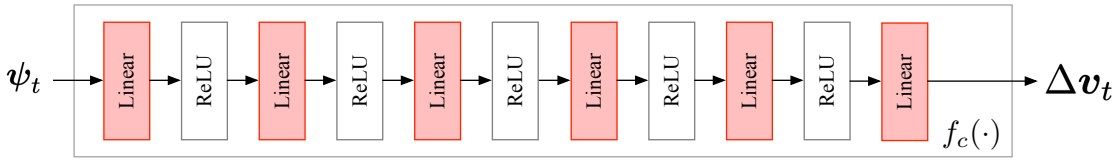

Figure 14. **Model Architecture for pose-dependent vertex displacement decoder $f_c(\cdot)$.**

References

[1] quartet. https://github.com/crawforddoran/quartet. 3

[2] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*, pages 561–578. Springer, 2016. 2

[3] Xu Cao and Takafumi Taketomi. Supernormal: Neural surface reconstruction via multi-view normal integration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20581–20590, 2024. 1

[4] Joao Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. Human pose estimation with iterative error feedback. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4733–4742, 2016. 1

[5] Liang-Chieh Chen. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 3

[6] Chen Guo, Tianjian Jiang, Xu Chen, Jie Song, and Otmar Hilliges. Vid2avatar: 3d avatar reconstruction from videos in the wild via self-supervised scene decomposition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12858–12868, 2023. 3

[7] Tong He, Yuanlu Xu, Shunsuke Saito, Stefano Soatto, and Tony Tung. Arch++: Animation-ready clothed human reconstruction revisited. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11046–11056, 2021. 3

[8] Sergey Ioffe. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 3

[9] Rawal Khirodkar, Timur Bagautdinov, Julieta Martinez, Su Zhaoen, Austin James, Peter Selednik, Stuart Anderson, and

Shunsuke Saito. Sapiens: Foundation for human vision models. In *European Conference on Computer Vision*, pages 206–228. Springer, 2025. 2, 3

[10] Peng Li, Wangguandong Zheng, Yuan Liu, Tao Yu, Yangguang Li, Xingqun Qi, Mengfei Li, Xiaowei Chi, Siyu Xia, Wei Xue, et al. Pshuman: Photorealistic single-view human reconstruction using cross-scale diffusion. *arXiv preprint arXiv:2409.10141*, 2024. 3, 4

[11] Zhe Li, Zerong Zheng, Lizhen Wang, and Yebin Liu. Animatable gaussians: Learning pose-dependent gaussian maps for high-fidelity human avatar modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19711–19722, 2024. 3

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

[13] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2304–2314, 2019. 3

[14] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 84–93, 2020. 3

[15] 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. 3

[16] Tianchang Shen, Jacob Munkberg, Jon Hasselgren, Kangxue Yin, Zian Wang, Wenzheng Chen, Zan Gojcic, Sanja Fidler, Nicholas Sharp, and Jun Gao. Flexible isosurface extraction for gradient-based mesh optimization. *ACM Trans. Graph.*, 42(4):37–1, 2023. 1

[17] Alex Trevithick, Matthew Chan, Michael Stengel, Eric Chan, Chao Liu, Zhiding Yu, Sameh Khamis, Ravi Ramamoorthi, and Koki Nagano. Real-time radiance fields for single-image portrait view synthesis. 2023. 3

[18] Yuliang Xiu, Jinlong Yang, Dimitrios Tzionas, and Michael J Black. Icon: Implicit clothed humans obtained from normals. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13286–13296. IEEE, 2022. 3

[19] Yuliang Xiu, Jinlong Yang, Xu Cao, Dimitrios Tzionas, and Michael J Black. Econ: Explicit clothed humans optimized via normal integration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 512–523, 2023. 2, 3, 4

[20] Yuliang Xiu, Yufei Ye, Zhen Liu, Dimitrios Tzionas, and Michael J Black. Puzzleavatar: Assembling 3d avatars from personal albums. *arXiv preprint arXiv:2405.14869*, 2024. 3

[21] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021. 3

[22] Zechuan Zhang, Zongxin Yang, and Yi Yang. Sifu: Side-view conditioned implicit function for real-world usable clothed human reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9936–9947, 2024. 3, 4