imGHUM: Implicit Generative Models of 3D Human Shape and Articulated Pose

Supplementary Material

Thiemo Alldieck* Hongyi Xu* Cristian Sminchisescu

Google Research

{alldieck,hongyixu,sminchisescu}@google.com

In this supplementary material, we detail our implementation and used architectures, show further results, discuss further ablations, and give more details on our experiments and comparisons. We also demonstrate how imGHUM can be used for differentiable rendering.

1. Implementation Details

In the following, we detail the implementation of imGHUM. We specify the used hyper-parameters and the architectures used in the ablation experiments. Finally, we give running times for imGHUM mesh extraction via Marching Cubes [5].

Hyper-parameters. We train imGHUM with a batch-size of 32, each of which contains 32 instances of $\boldsymbol{\alpha}$ paired with 512 on surface, 256 near surface, and 256 uniform samples for each instance. Our loss is composed as

$$L = \lambda_{o_1} L_{o_1} + \lambda_{o_2} L_{o_2} + \lambda_e L_e + \lambda_l L_l + \lambda_s L_s, \quad (1)$$

where L_{o_1} refers to the first part of L_o (distance) and L_{o_2} to the second part (gradient direction), respectively, and L_s refers to the semantics loss. We choose $\lambda_{o_1} = 1$, $\lambda_{o_2} = 1$, $\lambda_e = 0.1$, $\lambda_l = 0.1$, and $\lambda_s = 0.5$. Empirically we found that linearly increasing λ_{o_1} to 50 over 100K iterations leads to perceptually better results. We train imGHUM until convergence using the Adam optimizer [4] with a learning rate of 0.2×10^{-3} exponentially decaying by a factor of 0.9 over 100K iterations.

Architectures. The following architectures have been used for the baseline experiments: The single-part network has been used as described in the main paper totaling in 2.01M parameters. The deeper single-part network uses 10 instead of 8 layers, resulting in 2.53M parameters. The autoencoder is composed from a PointNet++ [6] encoder and our single-part decoder with a total number of parameters of 3.91M. The encoder consists of three PointNet++ set abstraction modules and two 512-dimensional fully-connected layers with ReLU activation.

Running Times. We extract meshes from imGHUM using using Octree sampling. Reconstructing a mesh in its bounding box and with a maximum grid resolution of 256^3 takes on average 1.08s using a NVIDIA Tesla V100. Hereby, the network query time sums up to 0.44s and Marching Cubes [5] (on CPU) takes 0.34s. The rest of the time is used by identifying the bounding box through probing (0.17s), Octree logic (0.05s), and transforming the samples to the part reference frames (0.07s). We query imGHUM in batches with a maximum batch-size of 64^3 samples, where one full batch takes on average 0.13s to compute. The resulting meshes feature approximately 100K vertices and 200K facets. Note that imGHUM allows creating meshes in arbitrary resolutions and can be queried and also rendered (*c.f.* §4.3) without generating an explicit mesh. For reference, we show imGHUM mesh reconstructions in different resolutions in fig. 1.

2. Results

In this supplemental material we show additional results for our application experiments (fig. 3, 4, 6, 7). Additionally, fig. 2 displays a large number of imGHUM instances with great variety in poses, shapes, hand poses, and facial expressions sampled from imGHUM's generative latent space. This demonstrates once more that imGHUM's level of detail, expressiveness and generative power is on par with state-of-the-art mesh-based models. Moreover, imGHUM can additionally be queried at arbitrary resolutions and spatial locations and models not only the surface, but also the space around the person.

3. Ablations

In this section, we report further results of our dataset ablation experiment and results of an additional ablation study on joint rotation parameterization.

* The first two authors contributed equally.

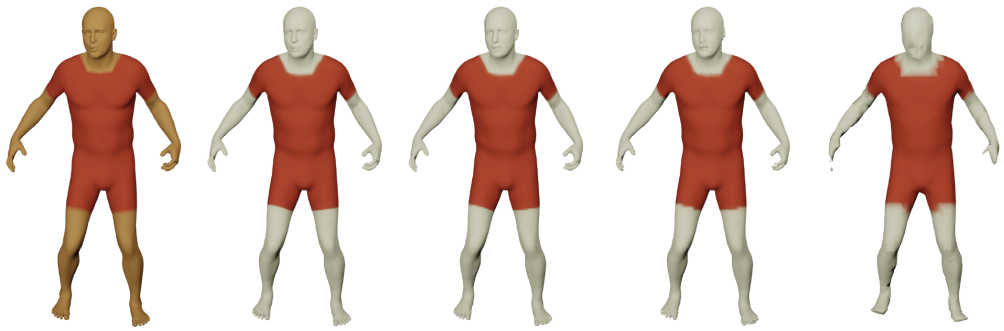

Figure 1. imGHUM mesh reconstructions in different resolutions. Left to right: ground-truth shape, 512^3, 256^3, 128^3, 64^3.

Model	IoU ↑	Chamfer $\times 10^{-3}$ ↓	NC ↑
Only scan registrations	0.901	0.091	0.975
imGHUM	**0.932**	**0.040**	**0.984**

Table 1. Numerical comparison of imGHUM trained with different data distributions evaluated on the registration test-set.

Model	IoU ↑	Chamfer $\times 10^{-3}$ ↓	NC ↑
Only scan registrations	0.834	2.561	0.942
imGHUM	**0.969**	**0.036**	**0.989**

Table 2. Numerical comparison of imGHUM trained with different data distributions evaluated on the GHUM samples test-set.

Model	IoU ↑	Chamfer $\times 10^{-3}$ ↓	NC ↑
6D	**0.969**	0.044	**0.989**
sin, cos	0.967	0.046	0.988
Euler	**0.969**	**0.036**	**0.989**

Table 3. Numerical comparison of imGHUM models using different representations for joint angles evaluated on the GHUM samples test-set.

Dataset. In the main paper we have shown that imGHUM benefits from being trained on both samples of GHUM and additionally on As-Conformal-As-Possible (ACAP) registrations of a corpus of human scans. While training only on scan data can represent the distribution of the scans well (tab. 1), it does not generalize sufficiently to poses that are not covered in this limited training set, as we show in tab. 2.

In fig. 5, we qualitatively show the effect of fine-tuning with scan data. Please note the increased level of detail in the faces and the enhanced soft-tissue deformation.

Rotation Representations. In tab. 3, we report metrics for imGHUM using different rotation representations for joint rotations $\boldsymbol{\theta}$. We have experimented with Euler angles, basic sin, cos Fourier mapping [7], and the recently proposed 6D representation [8]. Perhaps surprisingly, we found only minor differences in imGHUM's representational power using different rotation representations, both qualitatively and quantitatively. We, therefore, use Euler angles in this work as it is the most compact representation.

4. Applications

In the following, we explain the losses used in our triangle set surface reconstruction experiment, detail the residual model of the dressed and inclusive modeling experiment, and finally introduce another application namely pose estimation from silhouettes using differentiable rendering.

4.1. Triangle Set Surface Reconstruction

We describe our triangle set surface reconstruction experiment in the main paper (§3.3) and show more examples here in fig. 3. Our imGHUM reconstructions are performed under a weighted combination of losses as

$$\min_{\boldsymbol{\alpha}} L_o(\boldsymbol{\alpha}) + L_l^+(\boldsymbol{\alpha}) + L_l^-(\boldsymbol{\alpha}) \tag{2}$$

$$L_o(\boldsymbol{\alpha}) = \frac{1}{n}\sum_i |S(\hat{\mathbf{v}}_i, \boldsymbol{\alpha})| + \|\nabla_{\hat{\mathbf{v}}_i} S(\hat{\mathbf{v}}_i, \boldsymbol{\alpha}) - \hat{\mathbf{n}}_i\| \tag{3}$$

$$L_l^+(\boldsymbol{\alpha}) = \frac{1}{n}\sum_i \big(\phi(kS(\hat{\mathbf{v}}_i + \gamma_i \hat{\mathbf{n}}_i, \boldsymbol{\alpha})) - 1\big)^2 \tag{4}$$

$$L_l^-(\boldsymbol{\alpha}) = \frac{1}{n}\sum_i \big(\phi(kS(\hat{\mathbf{v}}_i - \gamma_i \hat{\mathbf{n}}_i, \boldsymbol{\alpha}))\big)^2, \tag{5}$$

where L_o is a surface sample loss (similar to eq. 2 in the main paper), and L_l^+, L_l^- are sign classification losses defined for points sampled along and opposite to the normals respectively ($\gamma_i \in [0, 0.05]$ is a Gaussian sampled distance).

Enabled by the implicit semantics of imGHUM, we can additionally exploit landmark losses as,

$$L_j(\boldsymbol{\alpha}) = \frac{1}{|M_j|}\sum_{i \in M_j} \left\|\mathbf{T}_i(\boldsymbol{\alpha})\mathbf{j}_i(\boldsymbol{\alpha}) - \mathbf{m}_{j,i}\right\|^2 \tag{6}$$

$$L_s(\boldsymbol{\alpha}) = \frac{1}{|M_s|}\sum_{i \in M_s} \left\|\mathbf{C}(\mathbf{m}_{s,i}, \boldsymbol{\alpha}) - \bar{\mathbf{m}}_{s,i}\right\|^2, \tag{7}$$

where $M_j = \{\mathbf{m}_j\}$, $M_s = \{\mathbf{m}_s\}$ are a collection of 3D landmarks defined over the joints and the surface, respectively. $\bar{\mathbf{m}}_s$ are the corresponding surface landmarks defined on the canonical mesh $\mathbf{X}(\boldsymbol{\alpha}_0)$. L_j aligns the transformed joint centers with the joint landmarks. The surface landmarks loss L_s queries the semantics for the ground-truth

Figure 2. Random imGHUM full-body and part instances sampled from imGHUM's generative latent codes. On the right, we show textured examples. Texturing and binary coloring is enabled by imGHUM's semantics.

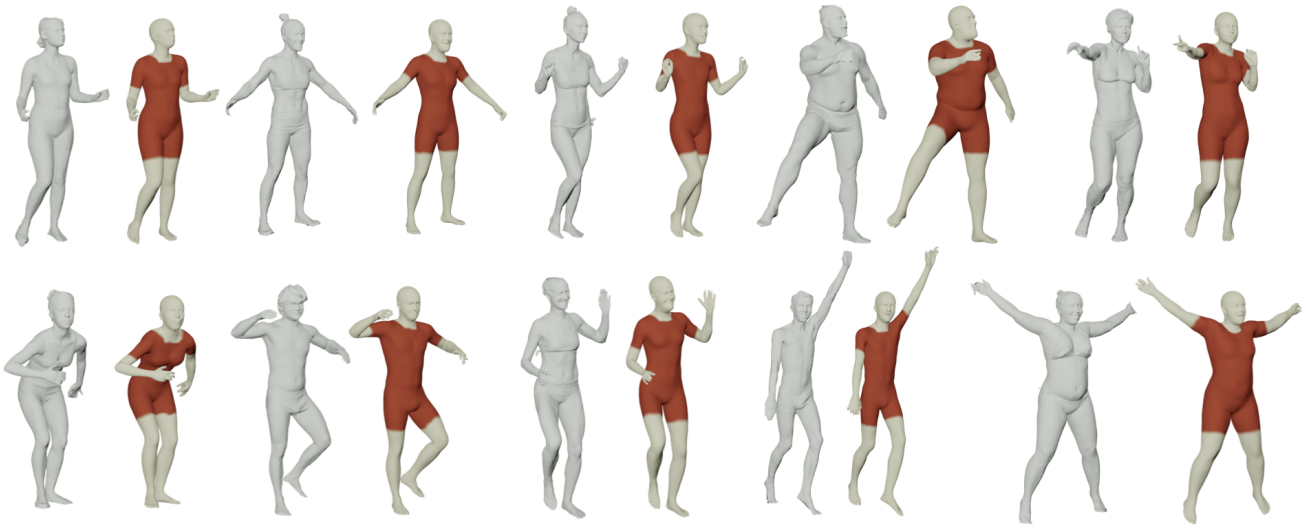

Figure 3. More examples for the triangle set surface reconstruction experiment. Each pair shows the ground truth scan (left) and our reconstruction (right). Notice the reconstructed facial expressions and hand poses.

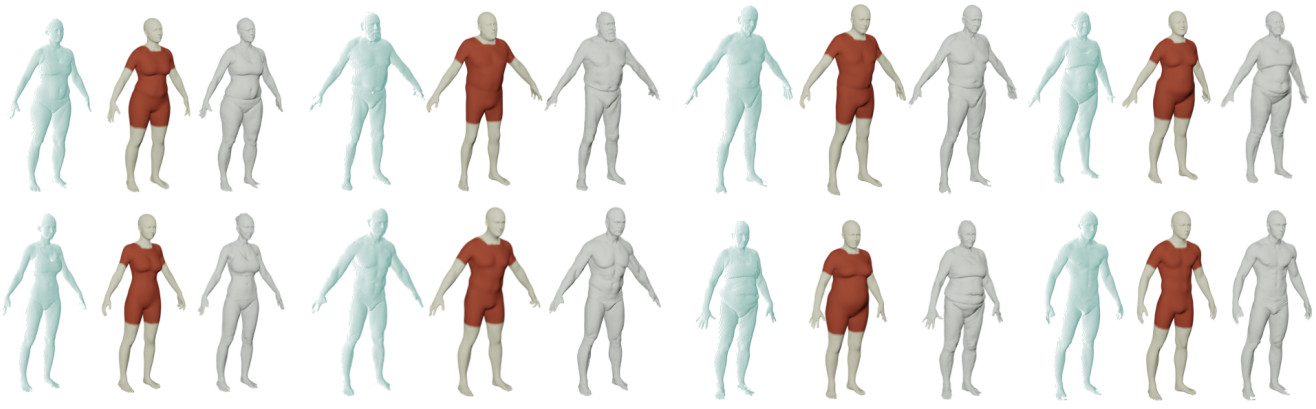

Figure 4. Remaining partial point cloud completion results. Left to right: input point cloud, imGHUM fit, and ground truth scan.

surface landmarks $\mathbf{m}_s$ conditioned on $\boldsymbol{\alpha}$. The semantics describe the position of the landmarks $\mathbf{m}_s$ w.r.t. the canonical mesh, and thus should match their correspondences $\bar{\mathbf{m}}_s$.

Given a triangle set mesh, one could also fit GHUM with landmarks and ICP losses. However, we note that imGHUM is not only able to perform equivalently on the landmark losses to mesh-based representations, but also exploits more information of the triangle set with its differential losses (eq. (2)) compared to ICP. The process of finding the nearest point for ICP at each optimization iteration is non-differentiable and the accuracy of the nearest point correspondences are highly sensitive to the initialization. In contrast, our imGHUM losses are fully differential everywhere and also exploit additional information encoded in the surface normals and the sign labels. Numerical comparisons are reported in §3.3 of the main paper.

4.2. Dressed and Inclusive Human Modeling

In the following, we detail our dressed and inclusive modeling experiment from the main paper. We also show more results in fig. 6. In order to learn a personalized shape of a given scan, we augment imGHUM with an MLP $\hat{S}$ consisting of four 256-dimensional layers. Each layer is followed by Swish nonlinear activation, and a skip connection is added to the middle layer. $\hat{S}$ modulates the signed distance field of the body to match the scan. These distance residuals could come from clothing, hair, other apparel items, or any divergence from the standard human template. We condition the output signed distance of the scan with both the distance and semantics fields of the body defined by imGHUM:

$$\hat{s} = \hat{S}(S(\mathbf{p}, \boldsymbol{\alpha})) = \hat{S}(s, \mathbf{c}). \tag{8}$$

We first fit imGHUM to the scan, similar to the trinagle set surface reconstruction experiment. Next, we train $\hat{S}$ on top of it. The training process is similar to imGHUM with the difference that we sample points from a single scan containing the desired personalizations. We only train the residual while keeping imGHUM fixed. We, therefore, have both the underlying human body and the personalized shape modeled separately as layers. We train a separate instance of $\hat{S}$ for each scan observation. Learning a combined model using an auto-decoder style learning scheme is possible but beyond the scope of this work.

We show two categories of personalizations: dressed humans and humans with limb differences. We compare imGHUM+residual with mesh-based GHUM ACAP registrations. In contrast to our template-free imGHUM+residual model, GHUM ACAP registrations have difficulties in explaining complex and layered structure and unsurprisingly fail entirely for large structural changes. We fit to scans of ten subjects with limb differences and 30 dressed human scans. Numerically, imGHUM+residual performs better than GHUM ACAP registrations with Chamfer distance 0.014×10^{-3} (ours, limb differences) / 0.018×10^{-3} (ours, dressed) versus 1.393×10^{-3} (GHUM ACAP, limb differences) / 0.021×10^{-3} (GHUM ACAP, dressed) and Normal Consistency 0.993 (ours, limb differences) / 0.990 (ours, dressed) versus 0.984 (GHUM ACAP, limb differences) / 0.976 (GHUM ACAP, dressed). imGHUM+residual is especially superior in explaining the scans of people with limb differences, due to large structural differences compared to the GHUM template mesh. Also qualitatively imGHUM+residual explains much more of the detail present in the input scans, see fig. 6 and fig. 7.

4.3. Differentiable Rendering

A benefit of imGHUM's SDF representation is the potential for rendering using sphere tracing [3]. During ray tracing the surface is located by stepping from the camera along a ray until a surface is passed. In sphere tracing the save step length is given by the current minimal distance to any point on the surface, i.e. the SDF value at the current location. For inexact SDFs, one can take a damped step to reduce the likelihood of over-shooting. Using this technique we can render among other things: imGHUM depth maps, normal maps, and semantics. Hereby, each pixel contains the last queried value of its corresponding camera ray. In the following, we compute differentiable binary silhouettes via sphere tracing and fit imGHUM to images using a silhouette alignment loss.

We implement differentiable approximate sphere tracing by taking a fixed number of steps. Concretely, we step $T = 15$ save steps into the SDF in the direction of each camera ray. At each final point $\mathbf{p}_T$ of each camera ray, we query the signed distance value and generate the binarized pixel as:

$$b = \frac{1}{\eta S(\mathbf{p}_T, \boldsymbol{\alpha})^2 + 1}, \tag{9}$$

with $\eta = 5000$ in our experiment. b is differentiable w.r.t. $\boldsymbol{\alpha}$ and thus can be used in optimization losses. We formulate a standard silhouette overlap loss and a sparse 2D joint landmark loss and use both to fit imGHUM to image evidence. Fig. 8 shows results of fitting imGHUM to image silhouettes.

5. Details on Compared Methods

As reported in the main paper, we change NASA [2] in contrast to their original version. Firstly, we train NASA based on the GHUM skeleton containing 63 parts. Originally, NASA was trained on SMPL containing only 24 parts. Another difference is the topology of GHUM. In contrast to SMPL, GHUM features an oral cavity that is also represented in our training data. Summarizing, we deploy NASA for a higher-dimensional model and thus a

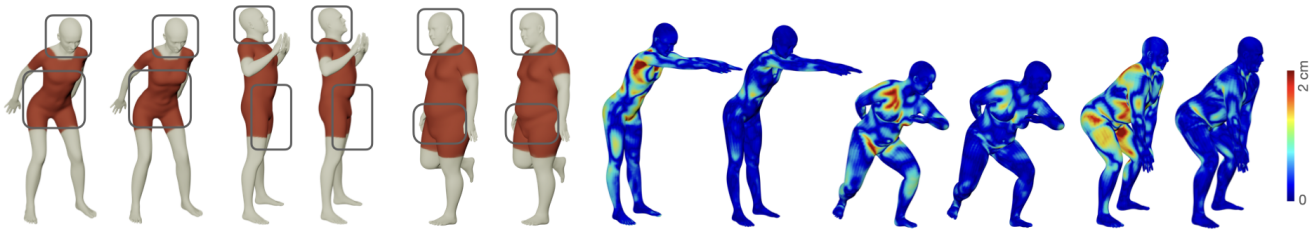

Figure 5. Added detail after fine-tuning on the registration dataset. We show imGHUM reconstructions before fine-tuning (left) and after fine-tuning (right) qualitatively and using error heat-maps (red means $\geq$ 2cm). Please pay attention to the faces, body shapes, and soft-tissue deformations (digital zoom in recommended).

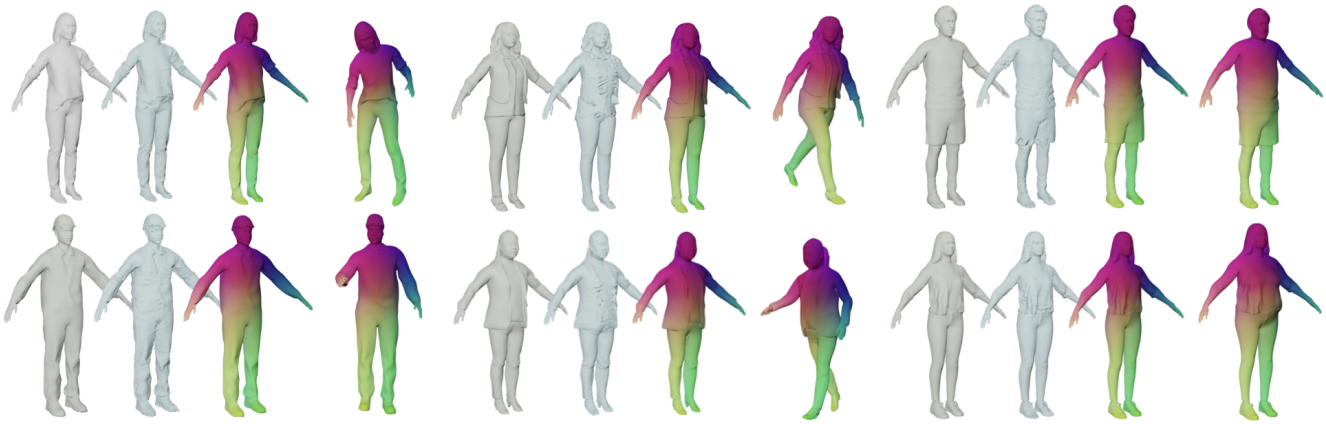

Figure 6. Dressed human modeling. From left to right: Scan, GHUM ACAP mesh registration, imGHUM+residual fit, reposed or reshaped imGHUM+residual. imGHUM+residual accurately explains all detail present in the input scan. GHUM ACAP mesh registrations have difficulties with complicated and layered structures. By changing the parameterization of the underlying imGHUM, we can repose and reshape the personalized models. The color-scale represents imGHUM semantics and thus correspondences between different instances.

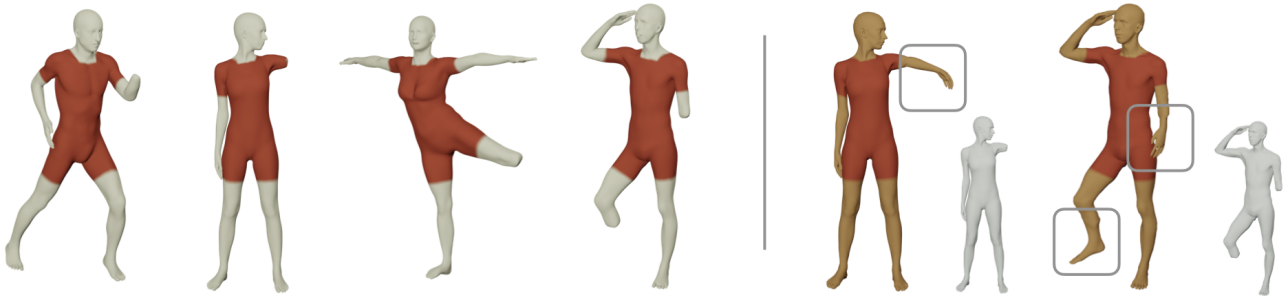

Figure 7. Inclusive human modeling. *Left:* imGHUM+residual can explain body shapes that do not match the standard template. *Right:* GHUM ACAP mesh registrations fail to explain these body shapes. For reference, we show ground truth scans in small. Missing limbs are deformed but still present.

Figure 8. Visual 3D reconstruction of imGHUM using differentiable silhouette and landmark losses. Left to right: image, observed silhouette, estimated silhouette, imGHUM reconstruction. By using a silhouette loss, we are able to accurately reconstruct body shapes.

harder task. For a fair comparison, we therefore use a larger and deeper architecture with eight 64-dimensional fully-connected layers for each part instead of the original four 40-dimensional layers. The new architecture features 1.92M parameters (original version has 0.38M) and has shown significantly better representation power. In contrast, we use a much smaller imGHUM architecture in this experiment. imGHUM has been originally designed to also explain shape variation and facial expressions. Since this experiment only features variation in pose, we can use a much smaller version. We use $2\times$ fewer layers in each part, each with half-dimensionality, resulting in only 0.64M parameters. This smaller-size imGHUM still performs significantly better than NASA in our experiments.

We have trained IF-Net [1] based on their original source code. Specifically, we use IF-Net for point clouds with 128^3 resolution featuring 2.6M parameters. We also follow their sampling and resizing strategy, such that the input point cloud always has a maximum side length of one unit. Finally, we train IF-Net task-specific (for full and partial point clouds), while we use the same imGHUM in all our comparisons.

References

[1] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *IEEE Conf. Comput. Vis. Pattern Recog.* IEEE, jun 2020. 6

[2] Boyang Deng, JP Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. Neural articulated shape approximation. In *Eur. Conf. Comput. Vis.* Springer, August 2020. 4

[3] John C Hart. Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces. *The Visual Computer*, 12(10):527–545, 1996. 4

[4] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *Int. Conf. Learn. Represent.*, 2015. 1

[5] Thomas Lewiner, Hélio Lopes, Antônio Wilson Vieira, and Geovan Tavares. Efficient implementation of marching cubes' cases with topological guarantees. *Journal of Graphics Tools*, 8(2):1–15, 2003. 1

[6] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Adv. Neural Inform. Process. Syst.*, pages 5099–5108, 2017. 1

[7] Andrei Zanfir, Eduard Gabriel Bazavan, Hongyi Xu, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu. Weakly supervised 3d human pose and shape reconstruction with normalizing flows. In *Eur. Conf. Comput. Vis.*, 2020. 2

[8] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5745–5753, 2019. 2