3D Segmentation of Humans in Point Clouds with Synthetic Data
Supplementary Material

Ayça Takmaz*[1], Jonas Schult*[2], Irem Kaftan†[1], Mertcan Akçay†[1], Bastian Leibe[2], Robert Sumner[1], Francis Engelmann[1,3] and Siyu Tang[1]

*,† indicate equal contribution

[1]ETH Zürich, Switzerland [2]RWTH Aachen University, Germany [3]ETH AI Center, Switzerland

Abstract

In this supplementary material, we provide further details about the synthetic data generation framework as well as the label acquisition process for real data. Furthermore, we describe our model architecture and our experimental procedures in more detail. Finally, we present additional quantitative and qualitative results.

1 Synthetic Data Generation Framework

In this section, we describe our framework for synthesizing virtual humans in realistic environments, and its use for obtaining synthetic training data with perfect ground truth for human instance and body-part segmentation tasks. Our pipeline consists of three main steps: (1) populating 3D indoor scenes (*illustrated in Fig. 1*), (2) rendering depth maps and label images from the 3D indoor scenes with synthetic humans, and (3) obtaining synthetic point clouds with ground truth labels. In the following, we provide details about each component of our pipeline.

1.1 Populating 3D Indoor Scenes

Real 3D Indoor Scenes. In this work, we use 3D real-world scenes from the ScanNet dataset [7], which is a large-scale 3D indoor dataset. The ScanNet [7] dataset features 1513 scenes and 707 rooms, and provides 3D surface reconstructions, 3D camera poses, captured RGB-D sequences, as well as annotations for segmentation tasks. We extend the ScanNet [7] dataset by generating synthetic humans in realistic poses, interacting with scenes from the dataset. Please note that there are several other available 3D indoor datasets such as [1, 3, 20], and our pipeline can be easily adapted to these datasets as well.

Scene Boundaries. The synthetic human generation method on which we base our approach, PLACE [22], requires the computation of scene boundaries as well as the signed distance field (SDF) for each input scene. Therefore, we first compute the SDF and scene boundaries for all training scenes in ScanNet [7]. The SDF value is 0 on the surfaces or boundaries of a set, which is utilized by PLACE to find suitable surfaces to place synthetic humans.

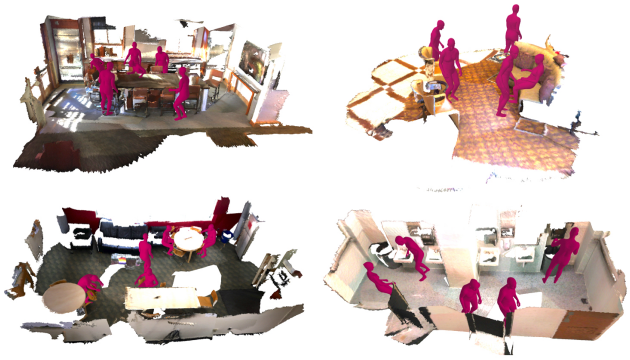


Figure 1: Synthetic Humans in ScanNet [7] Scenes. Example scenes (dining room, kitchen, living room, bathroom) populated with synthetic humans using PLACE [22] with instance-segmentation guided human location sampling.

PLACE: Proximity Learning of Articulation and Contact in 3D Environments [22]. For placing synthetic humans, we leverage PLACE [22], which is a generative human-scene interaction synthesis method. Given a 3D scene without humans, generation and placement of synthetic humans using PLACE [22] consists of several stages. First, a 3D cage within the scene is randomly sampled, and is transformed into the unit sphere for the computation of Basis Point Set (BPS) encoding of the scene, as well as the scene features. Then, a conditional variational autoencoder (cVAE) is utilized to generate body features conditioned on the scene features of the given 3D cage. Based on the scene BPS and the body features, a regressor is used to predict a

set of body mesh vertices, which are then transformed back to the original world coordinate system. In PLACE [22], the size of the 3D cage is chosen such that the cage is large enough to contain the full body mesh as well as the *nearby* scene objects. The 3D cage size is set as $2m^3$ following these constraints. Please see PLACE [22] for details.

Modified PLACE: Instance Segmentation Guided Human Location Sampling. PLACE [22] does not give full control over the interaction objects, which poses a limitation for our application as we are primarily interested in capturing humans in various poses with close human-scene interactions. Hence, we modify the PLACE [22] pipeline to address the need for selecting potential interaction objects and sample potential human locations, guided by object instance labels. In our modified pipeline, we use ground truth object instance labels from the ScanNet [7] dataset to identify areas in which the synthetic humans can closely interact with the scene. We identify the following object classes as suitable for our use case: *chair, couch, coffee table, seat, bed, table, bench, kitchen counter, sofa, dining table*.

The Human location sampling process in our modified pipeline consists of the following steps:

(1) For a given ScanNet scene, we first uniformly sample the number of humans, $n_{\text{humans}} \in [5, 10]$.

(2) Using the ground truth instance labels, we then identify n_{objects}, the number of object instances from the selected object categories present in the given scene.

(3) If $n_{\text{objects}} >= n_{\text{humans}}$, we uniformly sample a subset of the object instances to select n_{humans} objects. Otherwise, we select all available (n_{objects}) objects, and then randomly sample $n_{\text{random\_cage}} = n_{\text{humans}} - n_{\text{objects}}$ following the original implementation to reach the intended number of bounding boxes to place humans. We use the same 3D cage size ($2m^3$) as used for training PLACE [22].

(4) Using the selected bounding boxes, we follow the BPS encoding, scene feature extraction and human body synthesis stages from PLACE. We use 200 and 100 iterations for the simple and advanced optimization of PLACE, respectively. Moreover, we increase the weight of the collision loss term (from 8.0 to 10.0) in the advanced optimization to reduce inter-penetrations.

Overall, our pipeline enables us to generate humans in various poses while taking human-scene proximity into account for close interaction scenarios (with scene objects such as *tables* and *chairs*).

1.2 Rendering

We are primarily interested in creating a labeled synthetic dataset of partial point clouds obtained from depth scans. In order to obtain realistic depth maps and corresponding label images, we need to place a virtual camera in each scene with synthetic humans, and render frames using this virtual camera. With this purpose, we employ a simple virtual camera placement procedure.

First, we compute the scene center as the arithmetic mean of the global vertex coordinates of the full scene mesh. In order to better reflect the camera-to-ground distance of a potential handheld capture device (*e.g.* mobile phone, tablet), we uniformly sample a height value from the range $h_c \in [1.4, 1.6]$ m. We place the camera center at the scene center, and then apply a translation to ensure its z-coordinate is equal to the sampled height value h_c. Essentially, the camera is always aligned with the ground-plane, i.e., parallel to the xy-plane, however its height and viewing direction may change. We define the viewing direction as the rotation around the z-axis, and uniformly sample this rotation value within $[0°, 360°)$.

For any given scene with synthetic humans, we sample 40 frames – please note that one can arbitrarily increase the number of frames captured from a given 3D scene, and easily increase the scale of the dataset. At each rendering iteration, we re-sample the camera-to-ground distance and camera viewing direction. We render depth maps and label images with a resolution of 480×640 $(h \times w)$ with 60 degrees of horizontal FOV to imitate a Kinect depth sensor.

1.3 Kinect Depth Sensor Noise Simulation

In order to simulate Kinect depth sensor noise, we use SimKinect [9] – particularly the implementation available at [8]. For each depth image, we perform the noise simulation procedure using a scale factor of 100, baseline of 0.075 m, standard deviation of 0.5, filter size of 6, near-plane depth of 0.01 m and far-plane depth of 20 m. Noise simulation examples are shown in Fig. 2.

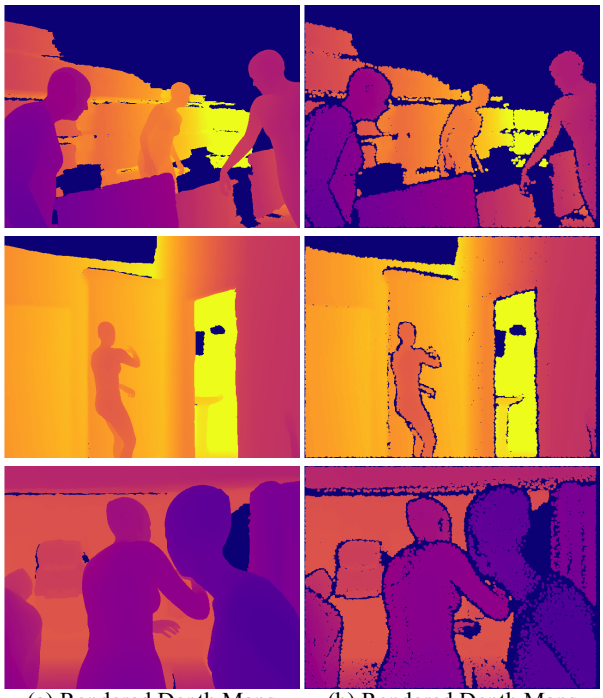
1.4 How many humans are there in each scene?

As described earlier in Sec. 1.1, the number of humans is uniformly sampled in $[5, 10]$ for each of the 1201 training and 312 validation scenes from the ScanNet [7] dataset. Since the rendering process captures only a portion of the 3D scene based on the sampled camera pose, the number of humans in each *rendered view* in the synthetic dataset is often smaller, and it varies in $[0, 8]$. In contrast, the EgoBody dataset [21] only has 2 humans per scene, and the BEHAVE [2] dataset only features 1 subject per scene. Please see Fig. 3 for an illustration of the number of human instances (per frame) vs. number of training samples.

In Fig. 4, we show example point clouds (obtained by back-projecting rendered depth maps) from our synthetic dataset, illustrating the varying number of human instances.

1.5 Merging Body Parts

In order to obtain body-part labels, we first map the faces of each SMPL-X [17] mesh to 26 body parts according to the mapping in [13]. Afterwards, we merge smaller body parts into larger ones as shown in Tab. 1 and Fig. 5, and



(a) Rendered Depth Maps (b) Rendered Depth Maps + *Kinect Noise Simulation*

Figure 2: Kinect depth sensor noise simulation. (a) Using the described rendering pipeline, depth maps are rendered from scenes populated with synthetic humans, (b) simulated Kinect depth sensor noise is applied to the rendered depth maps.

obtain 15 body part classes. We follow this merging scheme for all of our experiments (training and evaluation).

Merged Body Parts	Final Body Part
leftEye, rightEye, neck, head	head
leftToeBase, leftFoot	leftFoot
rightToeBase, rightFoot	rightFoot
leftHandIndex1, leftHand	leftHand
rightHandIndex1, rightHand	rightHand
spine, spine1, spine2, leftShoulder, rightShoulder	torso

Table 1: Merged Body Parts. Smaller body parts (*e.g.* eyes) were merged into larger ones (*e.g.* head)

1.6 Obtaining Labeled Synthetic Point Clouds

Rendered depth maps are backprojected to the 3D space, along with the label images to obtain perfectly labeled point clouds. After leveraging the depth images with simulated Kinect noise to backproject our label maps, we obtain partial point clouds which can occasionally be very sparse due to the virtual camera viewing direction as well as the simulated noise. Therefore, we perform a post-processing step to remove the scenes with less than 20k points. We use this pipeline to create a synthetic dataset for human semantic, human instance, and multi-human body-part segmentation tasks. For semantic and instance segmentation, we provide

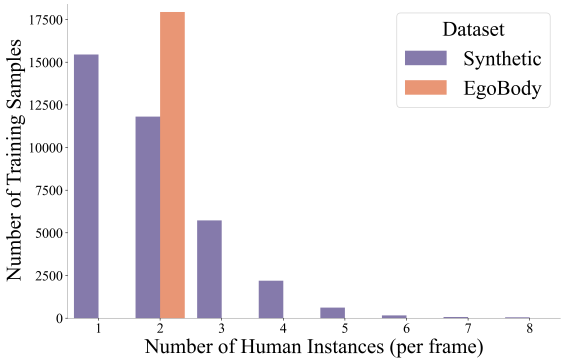


Figure 3: Number of human instances vs. number of training samples. Our synthetic dataset features scenes with up to 8 human instances whereas each EgoBody scene features exactly 2 subjects.

Figure 4: Example synthetic training scenes (point clouds). Our synthetic dataset features point clouds with a varying number of human instances.

two labels: background and human. For multi-human body-part segmentation, we map the faces of each SMPL-X [17] mesh to body-parts according to the mapping described in Sec. 1.5 and assign each point to one of the 15 body parts.

1.7 Dataset Size and Statistics

We place humans in 1201 training and 312 validation scenes from ScanNet, and render (capture) 40 frames per scene. Samples with fewer than 20k points are filtered out. Our final synthetic dataset consists of 36536 training and 12165 validation samples. For comparison, Real (EgoBody) dataset has 17943, and Real (BEHAVE) dataset has 41088 training samples.

2 Real Data Collection

In this section, we share details about our real data collection, processing and annotation pipelines.

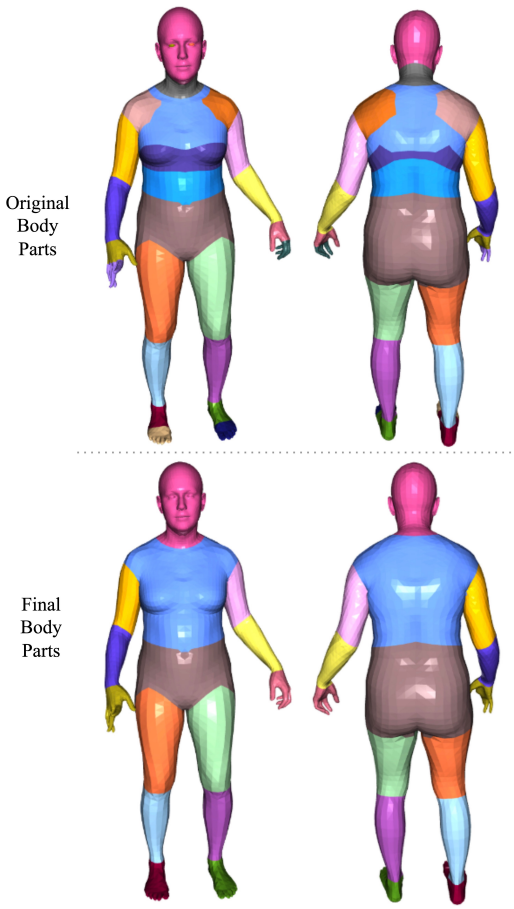


Figure 5: Illustration of body part merging. The first row shows original body parts, and the second row shows the body parts obtained after merging smaller parts into larger ones.

2.1 Pseudo Training Labels on Real Data

In this section, we give further details about our process for extracting pseudo ground truth labels for human semantic segmentation, instance segmentation and body part segmentation for the EgoBody [21] and BEHAVE [2] datasets. Our pipeline for extracting pseudo-labels is illustrated in Fig. 6 (left block).

EgoBody [21]. Each EgoBody scene features two subjects captured from multiple Kinect RGB-D cameras (3 or 5 cameras depending on the interaction sequence). Multi-view fitted SMPL-X [17] body parameters per each human are available. We process the frames at 1 FPS. We obtain the human instance masks by selecting scene points under 5 cm distance to the fitted body mesh. In order to obtain body-part segmentation labels, we first map the faces of each SMPL-X [17] body mesh to body parts according to the mapping in [13], and merge smaller body parts into larger ones. Then we assign each point in the human mask to the body part category of its closest neighbor in the fitted SMPL-X body mesh.

BEHAVE [2]. In each scene, there is one subject interacting with one object in a largely empty scene captured from 4 Kinect RGB-D cameras. Multi-view fitted SMPL [12] body model parameters are available. We obtain the human instance mask by selecting scene points under 5 cm distance to the fitted SMPL body mesh. As human point clouds were also released with the BEHAVE [2] dataset, we use these masks to refine the human instance masks we compute based on the distance between each point and its closest neighbor in the fitted body. In order to obtain body-part segmentation labels, we first map the faces of each SMPL [12] mesh to body parts according to the mapping in [13], resulting in 24 body parts (fewer than SMPL-X, where left-eye and right-eye are also specified as separate body parts), and merge the body parts (see Sec. 1.5). Then we assign each point in the human mask to the body part category of its closest neighbor in the fitted body mesh.

2.2 Manually Refined Evaluation Dataset

The EgoBody [21] dataset contains 125 interaction sequences captured by 3 or 5 Kinect cameras depending on the sequence. As the originally published train/validation/test splits were created based on separating first-person view subjects (the subject observed by the other subject wearing a head-mounted device) in each sequence, we created a new split such that none of the subjects overlap across splits. The split consists of 73 training sequences, 11 validation sequences, as well as 38 test sequences, while 3 sequences were removed to maintain a non-overlapping distribution of subjects across splits. From each of the test sequences, expert annotators have annotated 8 scenes, resulting in a test set consisting of 304 point clouds featuring a large variety of human poses, action types and occlusion levels. There is potential to expand the test set with a larger number of annotated test scenes in the future.

The annotation was performed using a 3D annotation tool [11]. The annotation tool is initialized with pseudo-labels for human instances. Then, the human instance masks are manually refined by annotators, as illustrated in Fig. 6 (right block, dotted line). Body part label refinement is guided by the resulting ground-truth human instance masks such that each point in the human mask is assigned to the closest body part in the original fitted body (please see Sec. 2.1), and each point outside of the refined human mask is removed from the body part mask.

2.3 Pseudo *vs.* Manually Refined Labels

Although the pseudo-ground truth labels for human masks and body parts were extracted using multi-view fitted body models from EgoBody, the labels can be noisy or incorrect in certain scenarios. Therefore, to obtain a more reliable evaluation set to conduct a thorough evaluation, we

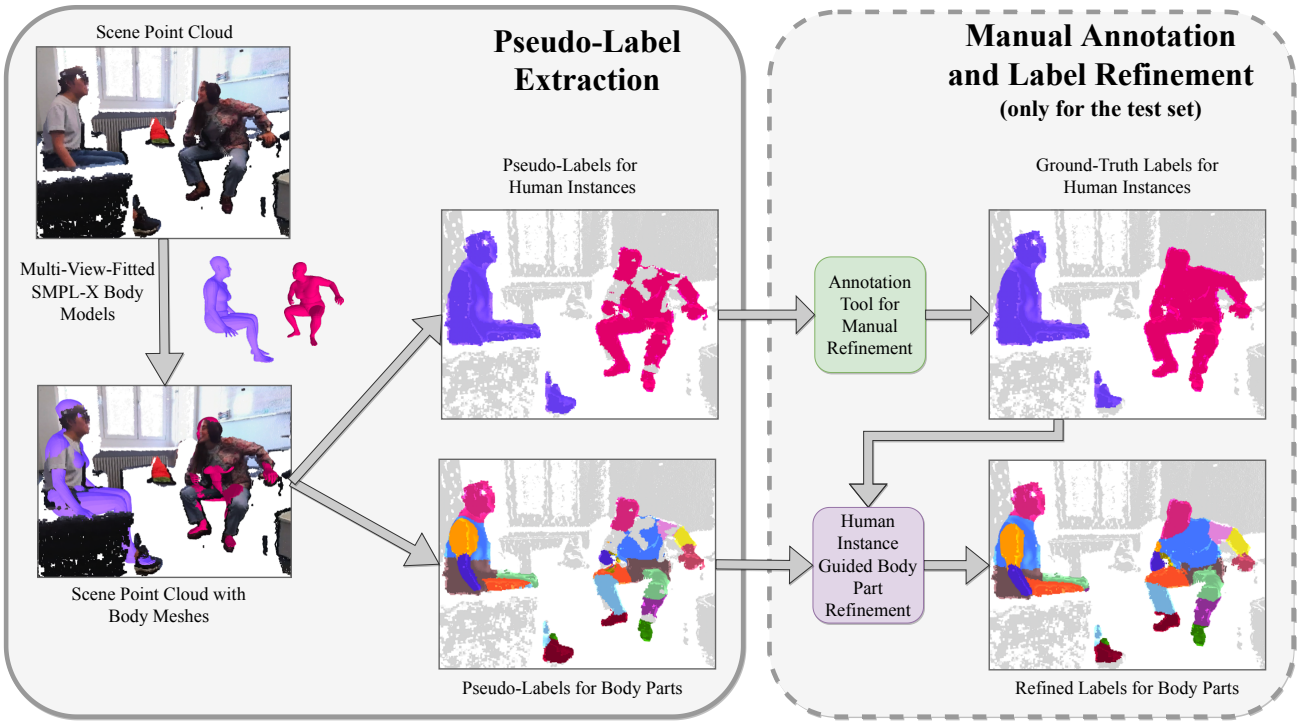


Figure 6: Pseudo-label extraction and label refinement. Pseudo-labels for human instances and body parts are obtained by performing the following procedure for each scene in EgoBody and BEHAVE: Each point in the point cloud obtained from a depth image is assigned to a human instance mask and a body part based on the distance between each point and its closest neighbor in the fitted body mesh. Only for the test set (EgoBody), expert annotators manually refine the human instance masks, which are then used to refine the body part labels.

refined the instance segmentation masks initialized by the fitted SMPL-X body meshes, following the annotation procedure described in Sec. 2.2. In Fig. 7, we illustrate the need for manual refinement, especially in case of close-contact interactions with scene objects (*e.g.* sitting on a sofa), loose clothing (*e.g.* wide-legged jeans) or unusual poses (causing a mismatch between the fitted body mesh and real human point cloud). Furthermore, we quantified the quality of the pseudo-ground truth labels by computing AP scores between the pseudo labels and the manually refined ground truth labels, resulting in AP^H : 91.9, AP^H_{50} : 99.3, and AP^H_{25} : 99.5, highlighting the need for manual annotations.

3 Human3D Architecture Details

We obtain strong multi-scale point features from a Minkowski Res16UNet18B [6]. We extract all 5 feature maps of sizes (256, 128, 128, 128, 128) from the U-Net decoder, pass them through a non-shared linear layer in order to project these point features to the transformer decoder features with 128 channels. Following Mask3D [18], we also use the modified transformer decoder of Mask2Former [5] instantiated with 8-headed attention and a feedforward network of 1024 dimensions. We sample point features for the cross-attention following Mask3D [18]. Human3D learns parametric human and body-part queries during training time. We assign 16 body-part queries to each of the 5 human queries. Following [16, 18], we use Fourier positional encodings based on normalized voxel positions. The full model, *i.e.* feature backbone and transformer decoder, uses 18.9 million parameters.

4 Experiments

In this section, we share further details about our experiments presented in the main paper, and provide additional results.

4.1 Clustering Details

For the semantic segmentation baselines KPConv [19] and MinkUNet [6], we obtain human instances by applying density-based clustering HDBSCAN [14,15] on the predicted human segments or body-part segments. We conduct a hyperparameter study to tune the parameters of the HDBSCAN algorithm, then we set HDBSCAN's minimum number of samples to 1200, and minimum cluster size to 1500. Each detected cluster of HDBSCAN represents a spatially contiguous instance. We assign each instance a confidence score of 100%.

4.2 Performance for Different Activity Types

We conduct an analysis to assess the effect of pretraining with synthetic data with respect to different human

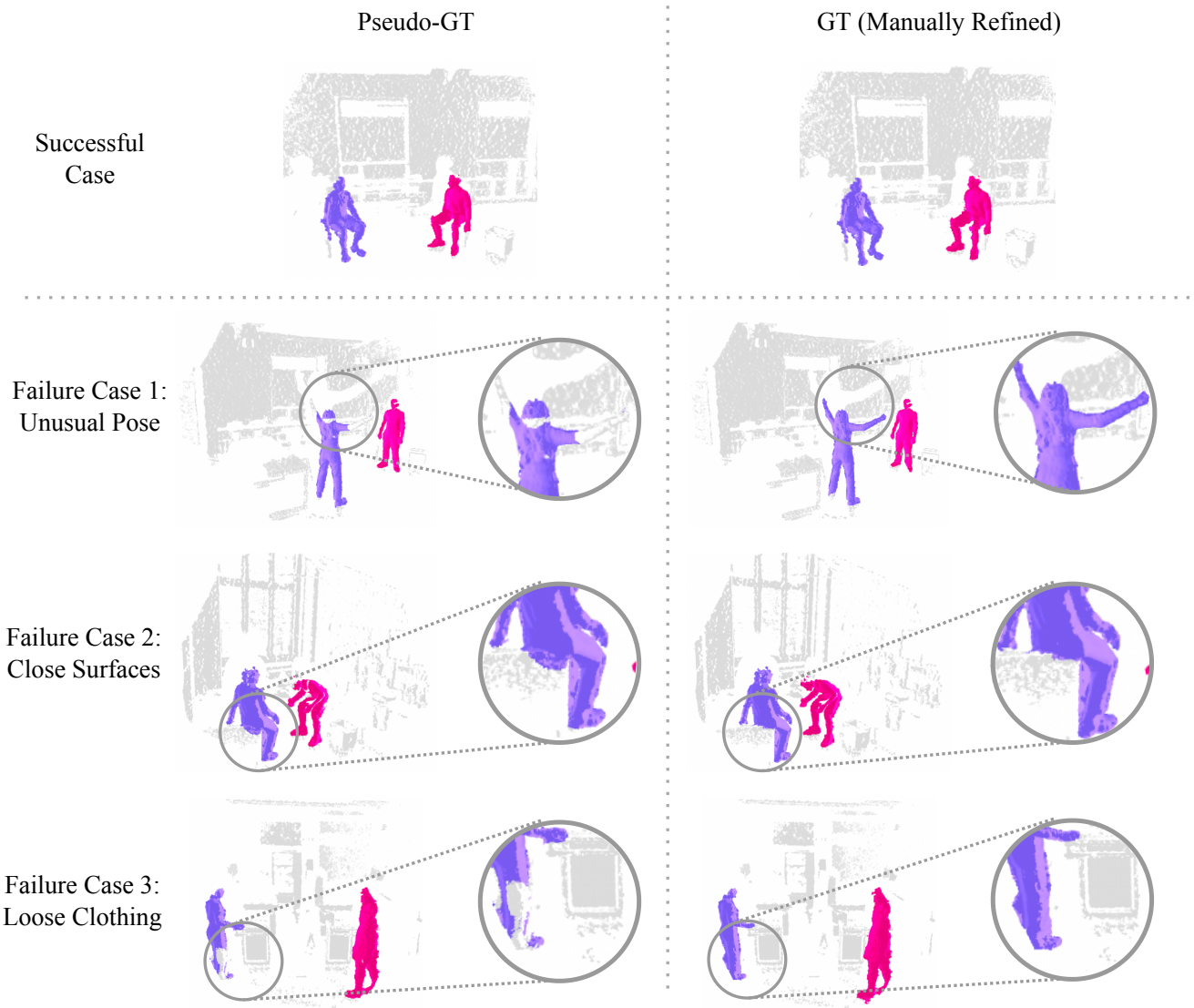


Figure 7: Pseudo-Ground Truth *vs.* Manually-Refined Ground Truth. Pseudo labels may fail particularly in the presence of (1) unusual poses, (2) nearby object- or scene-surfaces, and (3) loose clothing. Our manual annotations for human instances correct these failure cases (highlighted with circles), and provide an accurate and reliable evaluation benchmark.

activities. With this purpose, we create a set of activity categories as shown in Tab. 2, and manually annotate activities in each test scene. Please note that due to the nature of the dataset, our activity splits partly overlap. There are two main reasons for this. First, each EgoBody scene consists of two human subjects who potentially participate in different types of activities. In such cases, we assign the scene to both activity groups. Second, if subjects take part in compound activities (e.g. sitting down while pointing at an object on the table), we assign the scene to all relevant activity groups.

For each activity group we create, we report average precision scores for body parts (AP^P_{50}) with and without synthetic pre-training in Tab. 2. While pre-training with synthetic data results in consistent improvements on each activity category, we observe the largest improvement for actions that cause significant self-occlusions such as bending or walking.

Human3D	sit	stand	walk	sit down, stand up	lean, lie down	dance, exercise	kneel, bend	pick, put, hold an object	reach, touch, point at
w/o synth.	84.0	87.9	74.1	86.6	80.7	89.6	81.3	85.2	85.6
w/ synth.	**90.9**	**94.0**	**90.0**	**92.7**	**87.1**	**98.2**	**92.1**	**90.1**	**90.0**
	+6.9	+6.1	+15.9	+6.1	+6.4	+8.6	+10.8	+4.9	+4.4

Table 2: Multi-Human Body-Part Segmentation Performance for Different Activity Types. For each activity group, we report average precision scores for body parts (AP^P_{50}) with and without synthetic pre-training. We observe the largest improvement for actions that cause significant self-occlusions such as walking and bending.

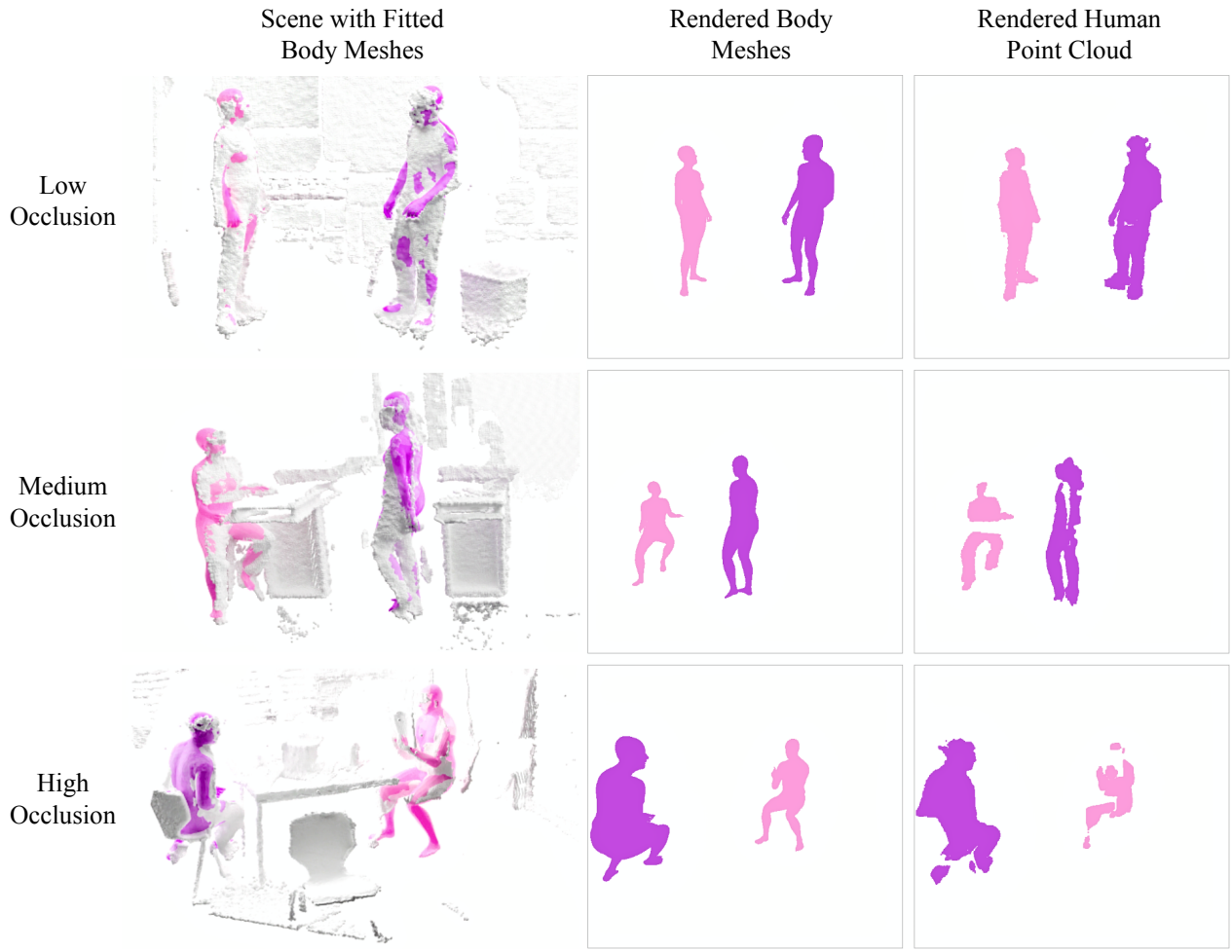


Figure 8: Occlusion Computation. In the first column, SMPL-X human body meshes fitted to humans in the EgoBody dataset are shown. The fitted body meshes for each human (*second* column) as well as human masks (*third* column) obtained from our manual annotation of the scene point clouds are projected onto an image. Using these rendered images, it is possible to compute an approximation of the occlusion level.

4.3 Occlusion Computation

In the main paper, we have shared our results from an analysis we conducted in order to assess the robustness of our model to occlusions. With this purpose, we split our test dataset into three groups based on the level of human occlusions: low (122 scenes), medium (104 scenes), high (78 scenes). For each scene in the EgoBody test set, we approximate the occlusion level of each human. To this end, we first project the fitted SMPL-X human body meshes for each human onto an image (see Fig. 8, *second* column). Then, we project the human masks obtained from our manual annotation of the scene point cloud (see Fig. 8, *third* column). Using these rendered images, it is possible to compute an approximation of the occlusion level. The occlusion level is inversely proportional to the ratio between the pixel-wise area of the human mask, and the area of the rendered body mesh. The computed ratio is only an *approximation* of the actual visibility, as the fitted body meshes are not perfect, and points are sometimes sparse in certain parts of the body due to Kinect depth noise. Each test scene consists of two human subjects, and we classify each scene based on the occlusion level of the highest occluded subject. Using this procedure, we first obtained an initial grouping based on the approximated visibility, which was then followed by a manual iteration to correct and account for potential mismatches between the fitted body and actual human mask.

4.4 Comparison to Image Baseline

Our approach is the first human segmentation method to operate directly on 3D point clouds of cluttered scenes. In the main paper, we compared our approach with two image-based baselines that operate on color images and project the segmentation masks onto the 3D point cloud obtained from the Kinect depth map.

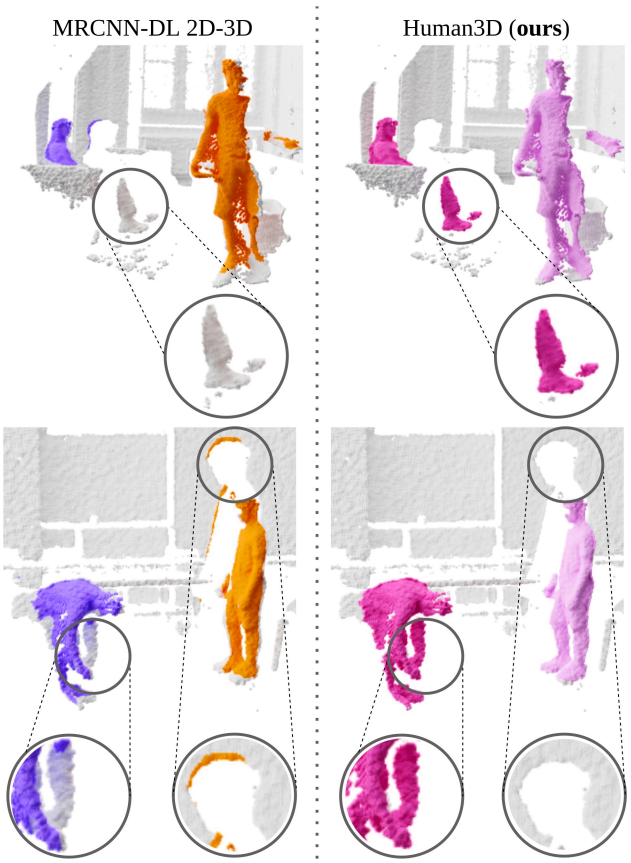


Figure 9: Failure cases of the 2D baseline. The first row shows a typical error of the Mask-RCNN baseline. The sofa occludes most of the human resulting in an incomplete human mask. In addition, the second example shows that small errors at the boundaries in 2D lead to incorrectly predicted 3D points projected far away.

In this section, we provide further implementation details about the *Mask-RCNN+DeepLabv3 2D-3D* baseline. This baseline closely follows the approach from [21]. The human semantic segmentation is obtained by applying a pretrained DeepLabv3 [4] to the Kinect RGB image. To obtain human instances, a pretrained Mask-RCNN is applied [10]. The final 2D human instance masks are then obtained by taking the intersection of the instance and semantic masks. These are then projected onto the 3D point cloud. The results are shown in Tab. 3. Both Mask3D [18] and our method Human3D outperform the baseline even without relying on color information, specifically on the AP^H metric which is more sensitive to inaccurate mask predictions. For both, we show the results of the models trained only on EgoBody as well as additionally pretrained on our synthetic data followed by finetuning on EgoBody, whereas the baseline is pretrained on much larger image datasets. The error cases are due to small mistakes in 2D at the boundary of a person which project to points far away in 3D. The baseline also has more difficulties to handle occlusions. Both scenarios show the advantage of directly operating on 3D data. We illustrate these cases in Fig. 9.

Model	Input	AP^H	AP^H_{50}
MRCNN-DL 2D-3D	RGB	61.3	97.3
Mask3D *(no pretraining)*	Geo. only	89.4	95.4
Mask3D *(pretrain.+finetune)*	Geo. only	95.6	98.7
Human3D *(no pretraining)*	Geo. only	90.5	95.2
Human3D *(pretrain.+finetune)*	Geo. only	**99.1**	**100**

Table 3: Comparison to image baseline. 3D instance segmentation scores on EgoBody test set. See also Tab. 3 in main paper.

5 Qualitative Results

EgoBody Test. In Fig. 10, we show additional qualitative results of Human3D on the EgoBody test set.

Synthetic Data Pre-Training. In Fig. 11 and Fig. 12, we qualitatively compare Human3D pre-trained on synthetic data with Human3D trained only on real EgoBody data. In Fig. 11, we observe that Human3D only trained on EgoBody data does not generalize to scenes with more than 2 individuals. The reason for this is that the EgoBody dataset only contains scenes with less than 3 people. When trained only on EgoBody, Human3D inevitably learns this bias and consequently fails on scenes with more than 2 people. In contrast, our synthetic dataset consists of scenes with up to 10 people. Human3D, pre-trained on synthetic data and fine-tuned on real EgoBody data, shows significantly better results for scenes with a larger number of people. In Fig. 12, we observe that pre-training with synthetic data provides robustness to occlusions and unusual poses, and results in improved multi-human body part segmentation predictions.

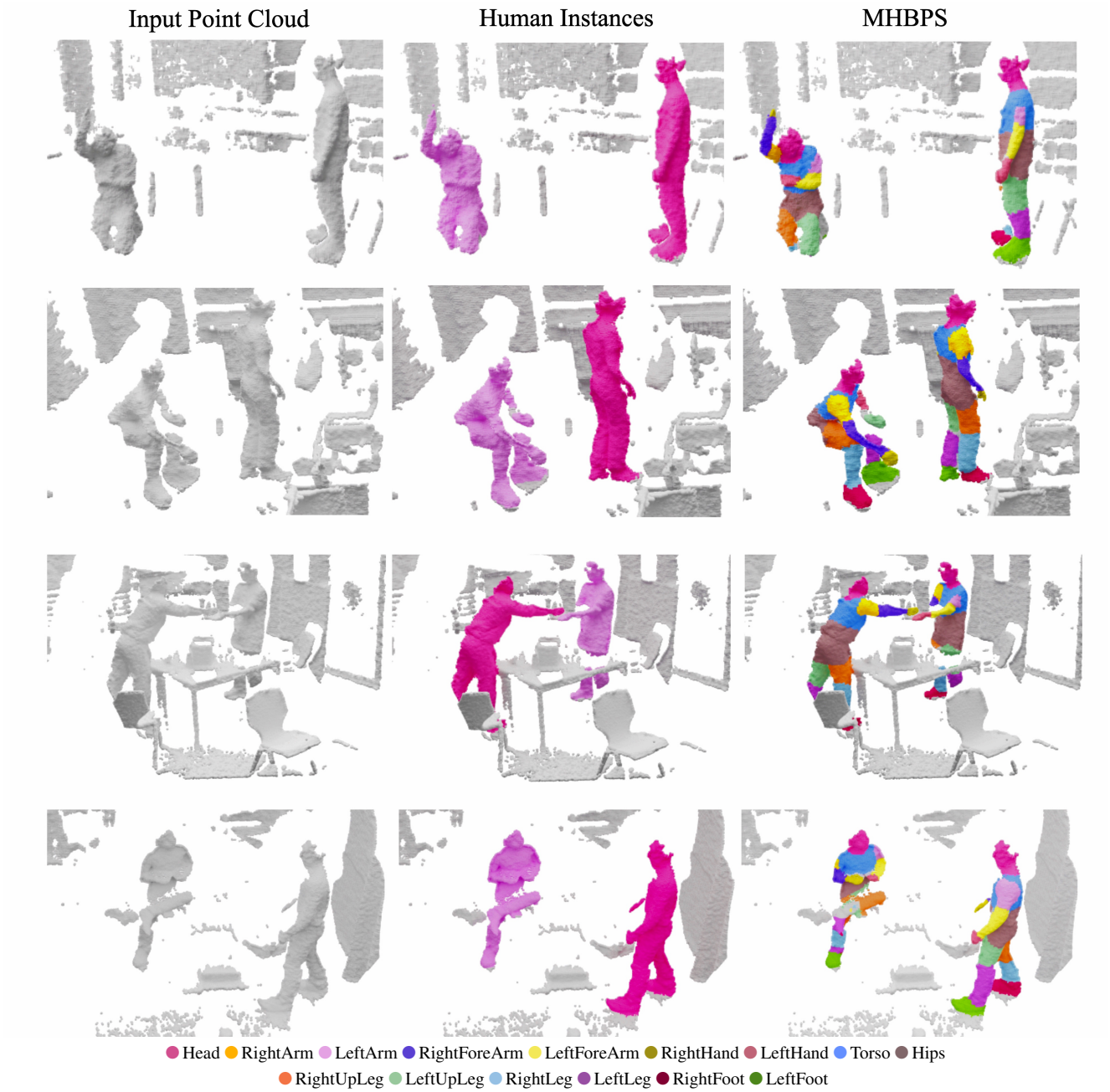


Figure 10: Qualitative Results on EgoBody Test Set. We show additional qualitative results of Human3D on the EgoBody test set. Human3D produces strong results even for humans in challenging poses, closely interacting or occluded by scene objects. The last row shows a failure case where Human3D predicts wrong body-parts for crossed legs.

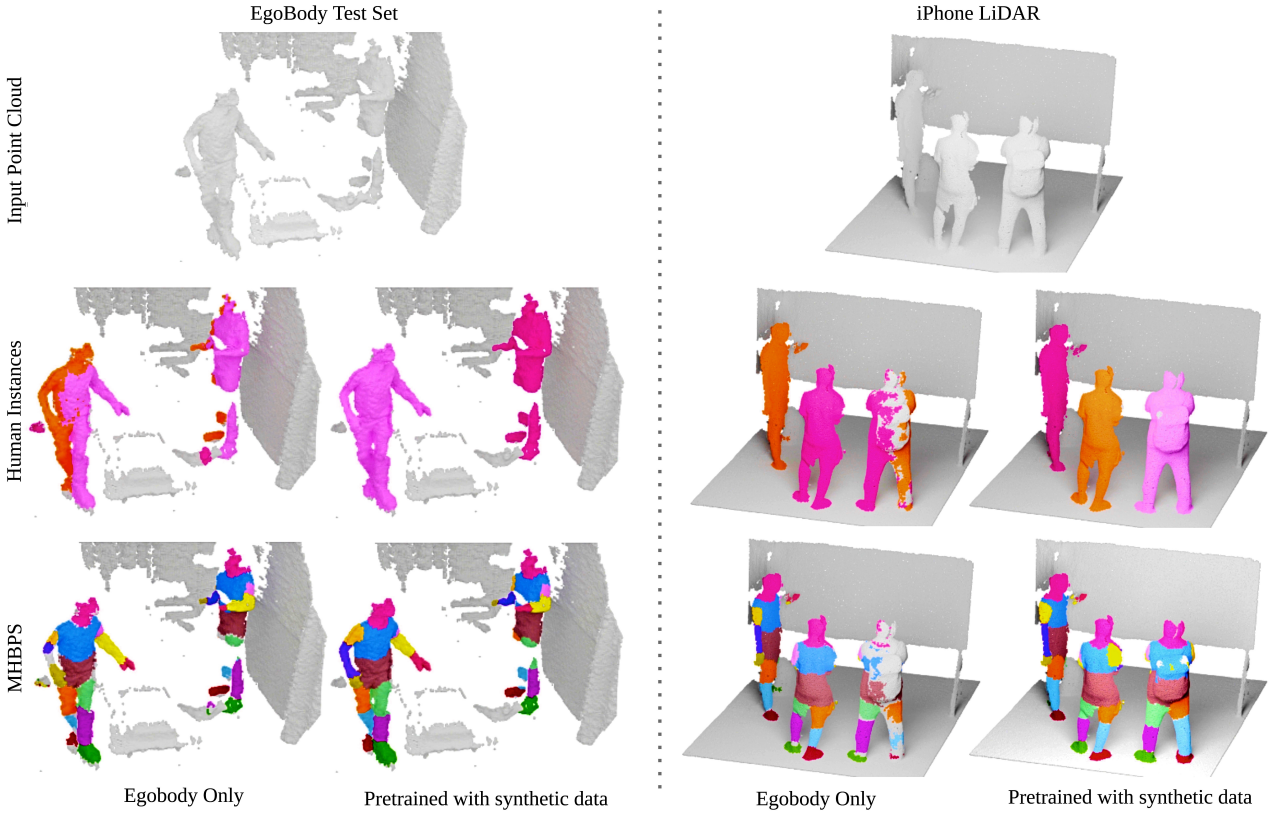


Figure 11: Pre-training with synthetic data improves upon training with EgoBody data only. In contrast to only training on real EgoBody data, Human3D pre-trained with synthetic data shows significantly better human instance predictions and even generalizes to scenes with more than 2 individuals.

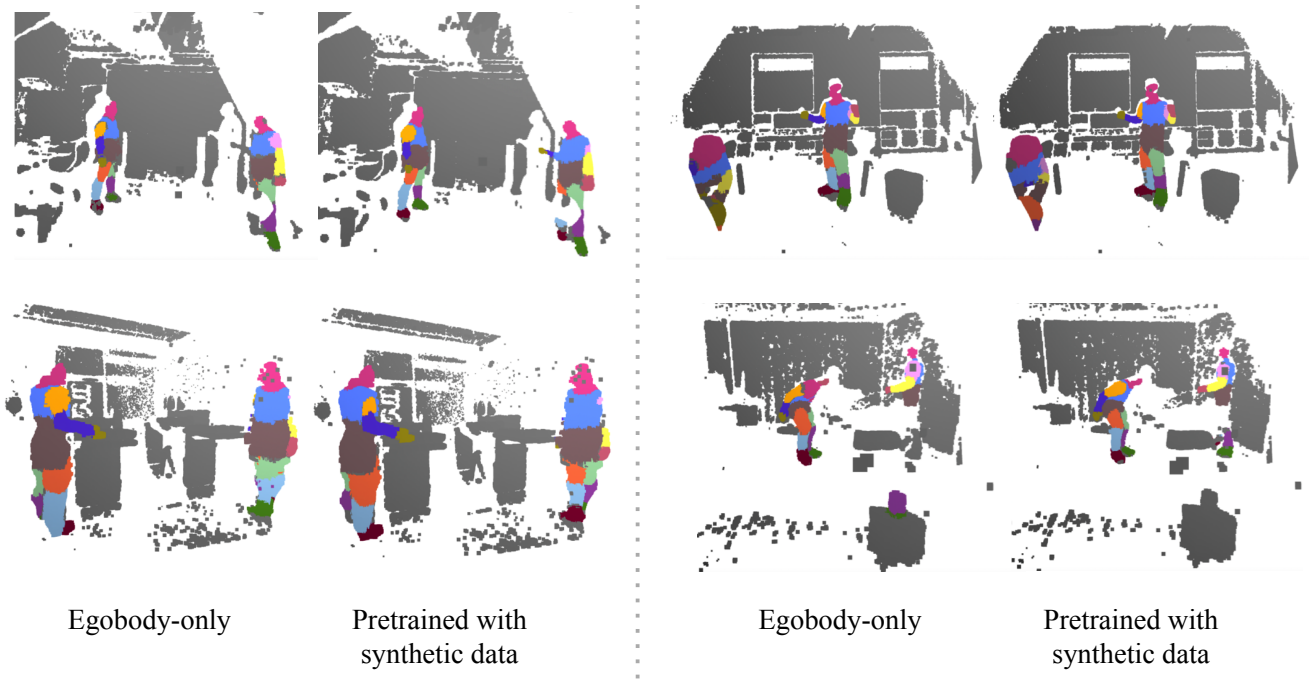


Figure 12: Pre-training with synthetic data improves upon training with EgoBody data only. Model only trained with EgoBody data often confuses body parts (e.g. left leg, right leg), and struggles in the presence of occlusions. In contrast to only training on real EgoBody data, Human3D pre-trained with synthetic data shows better body-part predictions on examples from the EgoBody test set.

References

[1] Iro Armeni, Ozan Sener, Amir R. Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3D Semantic Parsing of Large-Scale Indoor Spaces. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[2] Bharat Lal Bhatnagar, Xianghui Xie, Ilya Petrov, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. BEHAVE: Dataset and Method for Tracking Human Object Interactions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 4

[3] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D Data in Indoor Environments. In *International Conference on 3D Vision (3DV)*, 2017. 1

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. In *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2017. 8

[5] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention Mask Transformer for Universal Image Segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 5

[6] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 5

[7] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2

[8] Ankur Handa. Simulating kinect noise: adding noise to clean depth-maps rendered with a graphics engine. https://github.com/ankurhanda/simkinect. Accessed: 2022-11-16. 2

[9] Ankur Handa, Thomas Whelan, John McDonald, and Andrew J. Davison. A Benchmark for RGB-D Visual Odometry, 3D Reconstruction and SLAM. In *International Conference on Robotics and Automation (ICRA)*, 2014. 2

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *International Conference on Computer Vision (ICCV)*, 2017. 8

[11] Theodora Kontogianni, Ekin Çelikkan, Siyu Tang, and Konrad Schindler. Interactive Object Segmentation in 3D Point Clouds. In *International Conference on Robotics and Automation (ICRA)*, 2023. 4

[12] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A Skinned Multi-Person Linear Model. In *ACM Transactions On Graphics (TOG)*, 2015. 4

[13] Naureen Mahmood, Yehonal Azeroth, Sricharan Chiruvolu, and Denis Heid. Meshcapade Wiki. https://github.com/Meshcapade/wiki. Accessed: 2022-11-10. 2, 4

[14] Leland McInnes and John Healy. Accelerated Hierarchical Density Based Clustering. In *International Conference on Data Mining Workshops (ICDMW)*, 2017. 5

[15] Leland McInnes, John Healy, and Steve Astels. HDBSCAN: Hierarchical Density Based Clustering. *The Journal of Open Source Software*, 2017. 5

[16] Ishan Misra, Rohit Girdhar, and Armand Joulin. An End-to-End Transformer Model for 3D Object Detection. In *International Conference on Computer Vision (ICCV)*, 2021. 5

[17] 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 *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 3, 4

[18] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3D for 3D Semantic Instance Segmentation. In *International Conference on Robotics and Automation (ICRA)*, 2023. 5, 8

[19] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J. Guibas. KPConv: Flexible and Deformable Convolution for Point Clouds. In *International Conference on Computer Vision (ICCV)*, 2019. 5

[20] Johanna Wald, Armen Avetisyan, Nassir Navab, Federico Tombari, and Matthias Nießner. RIO: 3D Object Instance Re-Localization in Changing Indoor Environments. In *International Conference on Computer Vision (ICCV)*, 2019. 1

[21] Siwei Zhang, Qianli Ma, Yan Zhang, Zhiyin Qian, Taein Kwon, Marc Pollefeys, Federica Bogo, and Siyu Tang. EgoBody: Human Body Shape and Motion of Interacting People from Head-Mounted Devices. In *European Conference on Computer Vision (ECCV)*, 2022. 2, 4, 8

[22] Siwei Zhang, Yan Zhang, Qianli Ma, Michael J. Black, and Siyu Tang. PLACE: Proximity Learning of Articulation and Contact in 3D Environments. In *International Conference on 3D Vision (3DV)*, 2020. 1, 2