# 3D Segmentation of Humans in Point Clouds with Synthetic Data

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

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

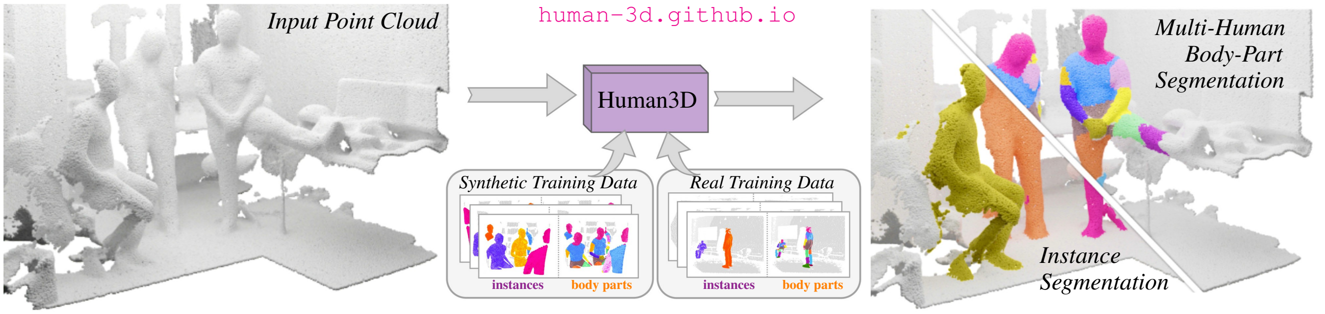


**Figure 1:** We propose Human3D, the first end-to-end model for 3D multi-human body-part segmentation in point clouds. Motivated by the lack of diverse and accurately labeled 3D human datasets, we generate synthetic training data of virtual humans in realistic 3D indoor scenes and demonstrate its potential in combination with pseudo-labels on real data. Above, we show an in-the-wild example of our model that is trained on synthetic data and real Kinect depth data, and tested on a reconstructed point cloud scanned with an iPhone LiDAR sensor.

## Abstract

*Segmenting humans in 3D indoor scenes has become increasingly important with the rise of human-centered robotics and AR/VR applications. To this end, we propose the task of joint 3D human semantic segmentation, instance segmentation and multi-human body-part segmentation. Few works have attempted to directly segment humans in cluttered 3D scenes, which is largely due to the lack of annotated training data of humans interacting with 3D scenes. We address this challenge and propose a framework for generating training data of synthetic humans interacting with real 3D scenes. Furthermore, we propose a novel transformer-based model, Human3D, which is the first end-to-end model for segmenting multiple human instances and their body-parts in a unified manner. The key advantage of our synthetic data generation framework is its ability to generate diverse and realistic human-scene interactions, with highly accurate ground truth. Our experiments show that pre-training on synthetic data improves performance on a wide variety of 3D human segmentation tasks. Finally, we demonstrate that Human3D outperforms even task-specific state-of-the-art 3D segmentation methods.*

## 1 Introduction

In this work, we address the task of segmenting humans in point clouds. In particular, we focus on 3D semantic segmentation (humans *vs.* background), 3D instance segmentation (masking multiple humans) and 3D multi-human body-part segmentation (segmenting human instances together with their body parts) as shown in Fig. 1 *(right)*.

As human-centered robotics and embodied AI are becoming more popular, there has been a growing interest in the development of methods for 2D human segmentation [11, 23, 25, 30, 83, 84, 89] and 3D human detection and segmentation [14, 37, 39, 66, 80]. While image-based methods have inherent limitations in their ability to reason in 3D, existing 3D methods mainly focus on simplified scenarios in which they only consider individual humans with pre-defined foreground segmentation masks and minimal occlusions. Real-life 3D scenarios, however, are typically cluttered, which can lead to strong occlusions when humans interact closely with each other and their environment.

3D segmentation of humans in point clouds (or depth maps) is a critical aspect of perceiving humans in various applications, such as AR/VR and robotics, in which depth sensors are commonly available and heavily used. For such applications, using point clouds has certain advantages. First, point clouds provide accurate scale and geometry, and are robust against illumination changes. Second, in the realm of human-related computer vision, point clouds are less biased towards visual appearance of humans. This can improve model fairness, and ensures better privacy when collecting data of real humans.

*,† indicate equal contribution.

Although there have been significant advancements in 3D scene understanding methods that operate directly on point clouds and segment indoor objects [15, 58, 65, 70], these advancements have not yet translated to the task of 3D *human* segmentation due to a lack of annotated humans in popular 3D indoor training datasets [1, 8, 16]. These indoor datasets usually lack diverse scenarios involving interactions between humans and cluttered real-world indoor environments. While outdoor datasets [4, 6] provide labels for pedestrians, they are limited in terms of human poses, actions, and occlusion patterns, making them less practical for indoor applications where humans closely interact with their surroundings. More recently, new datasets (BEHAVE [5], RICH [35], EgoBody [86]) provide depth recordings of humans interacting with their surroundings and other people. They are labeled with pseudo-ground truth human body meshes [50, 60] via multi-view registration processes relying on image segmentation and manual cleaning. To facilitate the labeling process, these datasets are often limited in terms of scene complexity, the number of people and poses, as well as occlusion and truncation patterns. Nevertheless, while tedious to annotate, these datasets can serve as realistic pseudo-labels for training 3D human segmentation tasks.

The key issue of recording and labeling real humans in complex indoor scenes is the time-consuming annotation process and thus its limited scalability. A promising alternative is *synthesizing* virtual humans as training data. Synthetic training data contains perfect labels that are impossible to annotate manually, and the creators have full control over dataset variation and diversity. Compared to generating color images, where it is challenging to render photorealistic humans [79], generating depth scans of 3D humans in 3D scenes is significantly easier, as the domain gap between real and synthetic point clouds is much smaller.

In this work, we describe a framework for synthesizing virtual humans in realistic environments, and show that it is possible to create synthetic training data that helps to improve 3D human segmentation in-the-wild. In addition, we propose a novel transformer-based model, called Human3D, that performs a wide variety of 3D human segmentation tasks in a unified manner. Human3D is the first model that directly addresses 3D multi-human body-part segmentation in point clouds of realistic environments. Human3D relies on a novel mechanism using *two-level* queries to jointly segment human instance masks and their associated body parts. Our experiments consistently demonstrate that pre-training models with synthetic data and fine-tuning with real data yields significant improvements over models trained exclusively on real data. Furthermore, our Human3D model trained for multi-human body-part segmentation achieves superior performance compared to task-specific state-of-the-art models for both 3D semantic and instance segmentation.

In summary, our contributions are as follows:

- Human3D, the first multi-human body-part segmentation model, that operates directly on real-world cluttered indoor 3D scenes.
- An approach for generating synthetic data of humans in 3D scenes and its use for synthesizing training data to improve 3D human segmentation.
- Manual annotation of 3D human instances on EgoBody [86] to evaluate human segmentation tasks.
- Extensive analysis showing the benefits of pre-training on synthetic data on multiple baselines and tasks.

## 2 Related work

**Multi-human parsing (MHP).** The goal of MHP is to segment multiple human instances along with their body parts. While well-explored in images [11, 23, 30, 83, 84, 89], it received less attention in point clouds. Several approaches [83, 84] are based on Mask R-CNN [30] which is one of the most effective methods for 2D instance segmentation. Yang *et al.* proposed RP R-CNN [83] which combines instance segmentation with semantics using a global semantics-enhanced feature pyramid network. While all of these methods require color images and cannot operate on purely geometric data such as point clouds, MHP and multi-human body-part segmentation in 3D are two very related tasks. As RP R-CNN [83] defines the state-of-the-art in MHP and is easily adaptable to our task, we consider RP R-CNN as a natural choice for a strong baseline.

**Segmenting humans in depth scans.** Several methods have been proposed for detecting humans [14, 80] and segmenting humans or body parts in depth scans [37,39,66,80]. Unlike ours, these methods often assume a given human segmentation mask, are limited to a single or few humans, and cannot handle strong occlusions. Instead, we focus on segmenting humans and body parts in real 3D scenes with multiple interacting people under strong occlusions.

**3D semantic and instance segmentation.** The goal of 3D semantic segmentation is to assign a semantic label to each point in a given 3D scene [1,2,15,21,22,24,33,34,36,41,45, 47, 49, 52, 61, 62, 68, 70, 75, 78, 81]. Instance segmentation further separates multiple objects within the same semantic class [13,19,20,26,32,38,42,44,65,73,76,82,85]. The field is largely driven by datasets [1, 8, 16] which ignore human labels, so these methods usually cannot segment humans. In this work, we train state-of-the-art methods KPConv [70], MinkowskiUNet [15], and Mask3D [65] on our proposed data, and compare them on different human segmentation tasks. Building on [15, 65], we propose the first end-to-end model for 3D multi-human body-part segmentation. In particular, the key idea of Human3D is to use *two-level* queries where the first level represents human masks and the second level represents their associated body parts.

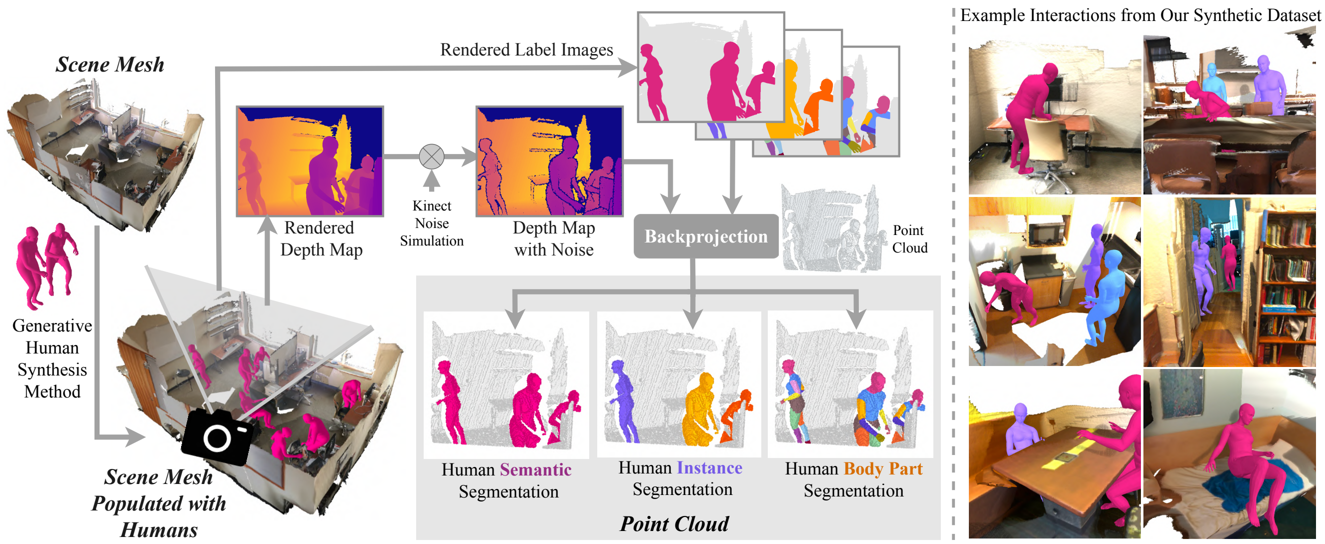

Scene Mesh
Rendered Label Images
Rendered Depth Map
Kinect Noise Simulation
Depth Map with Noise
Backprojection
Point Cloud
Generative Human Synthesis Method
Scene Mesh Populated with Humans
Human Semantic Segmentation
Human Instance Segmentation
Human Body Part Segmentation
Point Cloud
Example Interactions from Our Synthetic Dataset

**Figure 2: Synthesizing training scenes.** *Left:* Given a scene mesh from ScanNet [16], we populate it with *synthetic humans* based on PLACE [87]. We then render label maps and depth maps augmented with simulated Kinect noise [28]. Finally, the labels are backprojected to 3D using the synthesized depth maps to obtain highly accurate labels for human semantic, instance, and body-part segmentation. *Right:* Example interactions from our synthetic dataset featuring multiple humans, various occlusion levels and close contact with scene objects.

**Synthetic data generation.** Accurately annotating large amounts of data is tedious and occasionally not feasible, *e.g.* human body-part segmentation. This motivates an emerging trend towards synthesizing training data for various computer vision tasks [3, 18, 31, 48, 59, 63, 71, 72, 77, 79]. SURREAL [72] synthesizes 2D humans on top of real color images. However, the synthesized humans are not conditioned on the images, which results in unrealistic renderings. HSPACE [3] is a large-scale dataset of synthetic humans in synthetic indoor and outdoor environments, focusing on generating realistic color images. HUMANISE [77] is a language-conditioned human motion generator in 3D scenes and provides a dataset of synthetic, moving humans. Alternative methods [29,69,87,88] populate 3D scenes with synthetic humans. PLACE [87] synthesizes realistic 3D humans with natural poses conditioned on a given 3D scene. We extend PLACE to generate multiple 3D humans in ScanNet [16] scenes and condition the human generation to interactions with specific scene objects (*e.g.*, sofa, bed, chair).

## 3 Data Generation

In Sec. 3.1, we describe our framework for generating synthetic training data for human instance and body-part segmentation tasks. Then in Sec. 3.2, we describe our real data collection, processing and annotation pipelines.

### 3.1 Synthetic Training Data Generation

Fig. 2 illustrates our framework for generating synthetic training data. It populates real indoor scenes with synthetic humans and automatically generates labeled point clouds with perfect human and body-part labels that are otherwise difficult to obtain by manual labeling.

**Populating 3D indoor scenes.** We populate indoor 3D scenes from ScanNet [16], although our pipeline is suitable for other 3D indoor datasets as well [1, 8, 74]. To place synthetic humans in a given scene, we base our pipeline on PLACE [87], which is a generative human-scene interaction synthesis method. In order to obtain a large variety of human poses and close human-scene interactions, we modify [87] to perform instance segmentation-guided human placement. In our approach, we first identify object categories with which humans can naturally have close contact (*e.g.* chairs, tables), and use 3D object instance labels from ScanNet [16] to select these objects in the human-scene interaction synthesis process. We then sample potential interaction objects to generate up to 10 synthetic humans per scene, along with their SMPL-X [60] body parameters. The human synthesis approach is scene-aware as it encodes the nearby scene features. Our pipeline enables us to generate humans in various poses while taking human-scene proximity into account for close interactions. Further details about the human synthesis pipeline are in the sup. mat. Sec. 1.

**Rendering.** We render depth maps and label images from scene meshes we populate with humans. A virtual camera is placed at the scene center (arithmetic mean of the scene vertex coordinates), and its height is uniformly sampled from $[1.4, 1.6]\mathrm{m}$ to reflect the height of a potential handheld capture device (e.g. mobile phone, tablet). Camera viewing direction is always in parallel to the ground plane (xy-plane) and is rotated around the vertical axis by

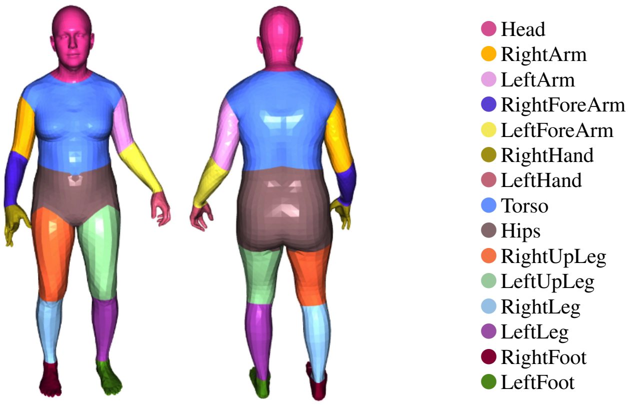


**Figure 3: Body-parts.** After merging smaller parts into larger ones (e.g. eyes into head), we obtain 15 body-part labels.

an amount uniformly sampled within $[0°, 360°)$. Rendered label images include annotations for semantics, instances, and multi-human body-parts (Fig. 2, *top*). We capture 40 frames per ScanNet scene, and re-sample the camera pose at each iteration. Further details about camera placement and sampling parameters are provided in the sup. mat. Sec. 1.2.

**Simulating Kinect noise.** We further refine the rendered depth maps by simulating Kinect noise using [28] to more closely mimic the depth data from a real Kinect sensor, as we use real Kinect data from EgoBody [86] for evaluation (Sec. 5.1). This allows us to combine real Kinect data (Sec. 3.2.1) and synthetic data for training. In preliminary studies, we found that simulating Kinect noise positively influences the segmentation quality. Please see sup. mat. Sec. 1.3 and Fig. 2 for further details and illustrations.

**Labeled point clouds.** The resulting depth maps and label images are back-projected into 3D space to obtain perfectly labeled partial point clouds. We use this pipeline to create a synthetic dataset for human semantic, instance, and multi-human body-part segmentation (MHBPS). For MHBPS, we map the faces of each SMPL-X [60] mesh to body-parts according to [54], then merge smaller parts into larger ones (*e.g.* eyes into head) and obtain 15 body-part classes. Resulting list of body parts is illustrated in Fig. 3. Please see the sup. mat. Sec. 1.5-1.6 and Tab. 1 for additional details.

## 3.2 Real Data Collection

### 3.2.1 Pseudo Training Labels on Real Data

Besides the synthetic data with perfect labels, we can also use real training data even though it requires expensive and time-consuming capturing processes and it produces less accurate, *i.e. pseudo,* labels. We use the recently released 3D human-scene interaction datasets EgoBody [86] and BEHAVE [5]. BEHAVE includes sequences of individual humans interacting with a single object in a mostly empty scene. EgoBody features social interactions between two humans captured in more cluttered static scenes. Both datasets provide multi-view depth recordings from several Kinect sensors, and carefully fitted SMPL [50] or SMPL-X [60] human body models. We obtain point clouds by back-projecting the Kinect depth to 3D and utilize the fitted body model parameters to obtain 3D human segmentation masks. We obtain body-part labels by selecting scene points within a fixed distance (5 cm) from the fitted body mesh, and assign each point to the closest body-part in the fitted body. Please refer to sup. mat. Sec. 2 for more details.

### 3.2.2 Manually Refined Evaluation Dataset

Pseudo-ground truth labels for human masks and body parts that were extracted using multi-view fitted body models from EgoBody (as described in Sec. 3.2.1) can be noisy in certain scenarios such as 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). As we cannot rely on noisy pseudo-labels for the evaluation of our model, we created a manually refined evaluation set based on the EgoBody dataset for a rigorous evaluation.

**Splits.** The EgoBody [86] dataset contains 125 interaction sequences captured by multiple Kinect cameras. As the original train/validation/test split was created with an aim to separate 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. Our split consists of 73 training sequences, 11 validation sequences, as well as 38 test sequences, while 3 sequences were removed to ensure a non-overlapping distribution of subjects across splits.

**Manual refinement.** For each of the selected 38 test sequences, expert annotators have annotated 8 scenes (point clouds), resulting in a test set consisting of 304 point clouds featuring a large variety of human poses, action types and occlusion levels. The annotation process is performed using a 3D annotation tool [40]. The labeling process is initialized with the noisy pseudo-labels for human instances based on the existing multi-view fitted human meshes (Sec. 3.2.1). Then, the human instance masks are manually refined by the annotators. Body part label refinement is then 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, and each point outside of the refined human mask is removed from the body part mask. Further details are in the sup. mat. Sec. 2.

# 4 Multi-Human Body-Part Segmentation

Our approach, Human3D, addresses the task of multi-human body-part segmentation (MHBPS) on 3D point clouds, *i.e.* it detects individual human instances and semantically partitions them into body-parts. Complex 3D indoor

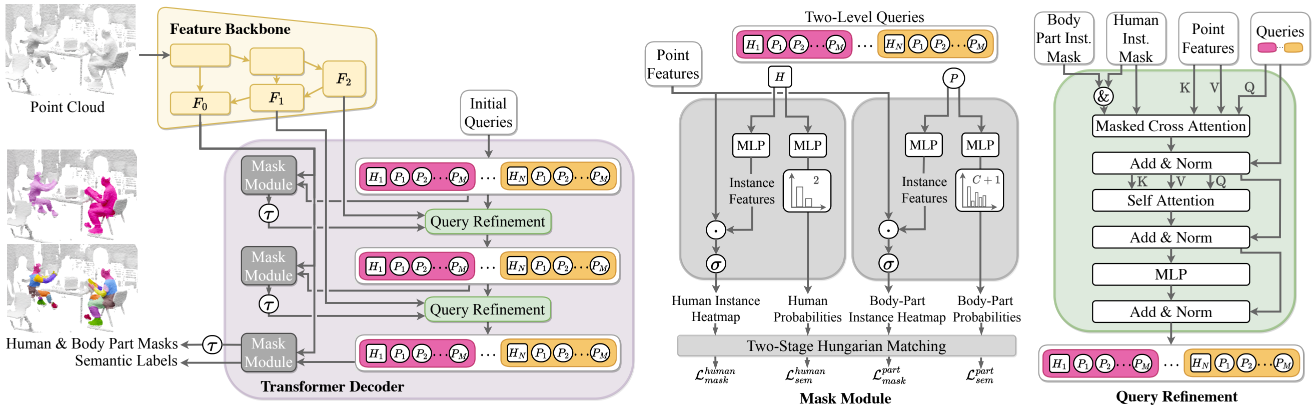


**Figure 4: Illustration of the Human3D model.** Our model consists of a sparse convolutional feature backbone and a transformer decoder *(left)*. The mask module jointly predicts human instance masks and body-part masks based on two-level queries *(middle)*, which are iteratively refined based on multi-scale point features within the predicted human instance mask *(right)*. $\boxed{H_i}$ represents human queries and $\textcircled{P_i}$ represents body-part queries. $\textcircled{\tau}$ applies a threshold of 0.5, $\textcircled{\sigma}$ is the sigmoid function and $\odot$ is the dot product operation.

environments, diverse human-object interactions, and close distances between humans make this task challenging. Not only is it required to correctly segment the body-parts, but it is also needed to correctly associate the body-parts with human instances. This needs capturing well-localized geometric details and high-level semantic context.

Inspired by the success of Mask3D [65] for 3D instance segmentation, we propose a transformer-based model with two dedicated query types: one for humans and one for body-part instances. We call these *two-level* queries. This key technical contribution enables the structured differentiation between human-level queries $\boxed{H_i}$ and body-part-level queries $\textcircled{P_i}$ (See Fig. 4). It is also essential to explicitly tie human masks together with their corresponding body-part masks during training such that body-part queries of one person are not supervised with ground truth masks of another person. Furthermore, we introduce a two-stage Hungarian matching mechanism, which guarantees that each ground truth human and body-part instance has a unique match with a predicted human instance and its associated body-parts. This matching explicitly enforces that human queries are tied to their respective body-part queries.

**Overview.** Our Human3D model is illustrated in Fig. 4. Our architecture consists of (1) a sparse convolutional feature backbone (🟨) implemented as a MinkowskiUNet [15], (2) a query refinement step (🟩) implemented as a masked transformer decoder (🟪) [12] which iteratively refines human and body-part queries by cross-attending to the multi-resolution hierarchy of the backbone decoder's point features $\{F_i\}_{i=0}^{2}$, and (3) a mask module (⬛) which predicts heatmaps for human and body-part instances together with their associated semantic class label.

**Human and Part Query Types.** The key technical contribution of this model, compared to prior work [65], is the two-level query types where each level specializes on one downstream task: The first level represents the human queries $H_1, ..., H_N$ (shown as $\boxed{H_i}$ in Fig. 4) which are trained to segment up to $N$ human instances in a scene. The second level represents the body-part queries $\{P_1^i, ..., P_M^i\}_{i=1}^{N}$ (shown as $\textcircled{P_i}$ in Fig. 4). To each one of the $N$ human queries, we associate $M$ body-part queries. This explicit modelling of correspondences between $M$ body-part queries and a single corresponding human query, results in two important properties: (1) We can directly extract the body-part segmentation for each human instance and (2) during query refinement (Fig. 4, 🟩), we enable information flow between human instances and body-parts via self-attention among human and body-part queries. We therefore update human instance masks based on their predicted body-part masks, and vice versa. Further, we tie body-parts to their associated human instance, by restricting body-parts to only cross-attend to backbone point features which lie within the corresponding human mask (Fig. 4 $\textcircled{\&}$, *right*).

**Two-Stage Hungarian Matching.** Human3D infers $N$ human instances and $N \cdot M$ body-parts during a single feed forward pass of the model. As these predictions as well as the ground truth targets are unordered, we need to find optimal correspondences between these two sets in order to optimize the model. Typically, the Hungarian Algorithm [43] is deployed to find such optimal correspondences [7,12,65]. However, for MHBPS we cannot simply match human and body-parts independently. We additionally have to guarantee that both the predicted body-part masks and the human mask are mapped to target body-part masks and target human mask of the same human. We therefore introduce a two-stage Hungarian matching approach:

In the first stage, we define the assignment cost for a predicted *human* instance $h$ and a target instance $\hat{h}$ as follows:

$$\mathcal{C}_1(h, \hat{h}) = \mathcal{L}^{\text{human}}_{\text{mask}}(h, \hat{h}) + \mathcal{L}^{\text{human}}_{\text{sem}}(h, \hat{h}) \qquad (1)$$

```python
def two_stage_matching(h_mask, h_prob, p_mask,
                       p_prob, h_gt, p_gt):
  # human-level: h_mask, h_prob and GT h_gt
  # part-level: p_mask, p_prob and GT p_gt

  # 1-stage: human-level predictions <-> GT
  h_indx, loss = Hungarian(h_mask, h_prob, h_gt)
  L_total = loss

  # for each (pred, gt) matched human instance
  for (pred_i, gt_j) in h_indx:
    mask = p_mask[pred_i]
    prob = p_prob[pred_i]
    gt = p_gt[gt_j]

    # 2-stage: part-level predictions <-> GT
    _, p_loss = Hungarian(mask, prob, gt)

    L_total += p_loss
  return L_total
```

**Listing 1:** Two-Stage Hungarian Matching Algorithm.

The cost for matching human *masks* is a weighted combination of the Dice loss [17] and binary cross-entropy $\mathcal{L}^{\text{human}}_{\text{mask}} = \lambda_{\text{BCE}}\mathcal{L}_{\text{BCE}} + \lambda_{\text{dice}}\mathcal{L}_{\text{dice}}$ while the semantic classificaton loss is defined as $\mathcal{L}^{\text{human}}_{\text{sem}}=\lambda_{\text{cl}}\mathcal{L}_{\text{CE}_{\text{cl}}}$. Using the Hungarian Algorithm [43], we find a globally optimal assignment between predicted and ground-truth human instances. Following [7], we represent this assignment by a permutation $\sigma \in \mathfrak{S}_N$ which maps the target human instance $\hat{h}^j$ to the predicted human instance $h^{\sigma(j)}$. We then use this optimal assignment between human masks to match their corresponding *body-parts* $p$ using the following cost matrix:

$$\mathcal{C}_2(p^{\sigma(j)}, \hat{p}^j) = \mathcal{L}^{\text{part}}_{\text{mask}}(p^{\sigma(j)}, \hat{p}^j) + \mathcal{L}^{\text{part}}_{\text{sem}}(p^{\sigma(j)}, \hat{p}^j) \tag{2}$$

$\mathcal{L}^{\text{part}}_{\text{mask}}$ and $\mathcal{L}^{\text{part}}_{\text{sem}}$ are analogously defined to their human instance counterparts $\mathcal{L}^{\text{human}}_{\text{mask}}$ and $\mathcal{L}^{\text{human}}_{\text{sem}}$. After establishing correspondences between human masks and their corresponding body-parts, we optimize all auxiliary predictions after each of the $L$ query refinement steps:

$$\mathcal{L} = \Sigma_l^L\ \mathcal{L}^{\text{human},l}_{\text{mask}} + \mathcal{L}^{\text{human},l}_{\text{sem}} + \mathcal{L}^{\text{part},l}_{\text{mask}} + \mathcal{L}^{\text{part},l}_{\text{sem}} \tag{3}$$

This loss enforces that human masks as well as their body-part masks are matched to the same ground truth human.

We provide an outline of the Two-Stage Hungarian Matching algorithm in Listing 1.

**Extracting body-part segmentations.** Human3D represents body-parts as *instances*. We therefore now describe how we merge these body-part instances to obtain a semantic body-part segmentation for each human instance. First, we restrict body-parts to lie within their corresponding human instance masks, *i.e.* points of body-parts outside the human mask are set to background. Second, for each point in the human mask, we obtain the semantic body-part label of the body-part instance mask with the highest confidence. If the highest confidence is below 10%, we ignore the prediction and assign the point to background.

# 5 Experiments

In this section, we first compare our Human3D model with state-of-the-art segmentation methods for 3D point clouds and 2D images (Sec. 5.1). We then provide analysis experiments on occlusions, an ablation study of Human3D and demonstrate the benefits of pre-training with synthetic data (Sec. 5.2). Finally, we show qualitative results of our approach (Sec. 5.3). Additional analysis is provided in the supplementary material Sec. 4 and Sec. 5.

## 5.1. Comparing with State-of-the-Art Methods

**Dataset and Test Annotations.** We train on our synthetic data with perfect labels (Sec. 3.1), and on real data with pseudo labels (Sec. 3.2.1). For a rigorous evaluation, we further require accurate per-point ground truth labels since we cannot rely on the noisy pseudo-labels. As no such dataset exists, we *contribute* new annotations based on EgoBody (please see Sec 3.2.2). We define a test split such that there is no overlap of human subjects with the training set. The labeling process is initialized with the noisy pseudo-labels based on the existing multi-view fitted human meshes [86]. Expert annotators then manually label the test scenes using an interactive point cloud labeling tool [40] to refine the noisy instance masks (illustrated in supplementary material Fig. 6-7). For body-part labels, pseudo-ground truth labels are refined using the manually corrected instance masks. The test set contains 304 point clouds and 608 humans with various poses, actions, and occlusions.

**Tasks and Metrics.** We evaluate our approach on three different 3D point cloud tasks: human/parts semantic segmentation, human instance segmentation and multi-human body-part segmentation (MHBPS). For *human* semantic segmentation and *body-part* semantic segmentation, we report the mean intersection-over-union (denoted as mIoU$^H$ and mIoU$^P$). For instance segmentation, we use the average precision (AP). We denote human instance segmentation scores as AP$^H$, and multi-human body-part segmentation scores (MHBPS) as AP$^P$. For MHBPS, we additionally report the percentage of correctly parsed body parts (PCP) used by the 2D multi-human parsing community [46]. Metrics are evaluated at overlaps of 25%, 50%, and averaged over the range [0.5:0.95:0.05] as in ScanNet [16].

**Human3D Training Details.** For pre-training and fine-tuning, we train Human3D for 36 epochs each. We optimize the network with AdamW [51] and a one-cycle learning rate scheduler [67] with a maximal learning rate of $10^{-4}$ and a batch size of 4 scenes. Data augmentation includes horizontal flipping, random rotations around the z-axis, elastic distortion [64], and random scaling by Uniform[0.9, 1.1]. Training (including pre-training and fine-tuning) with 2 cm voxels takes 5 days on a single NVIDIA RTX 3090 GPU.

**Methods in Comparison.** We compare with a wide range

| | | *3D Multi-Human Body-Part Segmentation* | | | | | | *3D Instance Seg.* | | | *3D Semantic Seg.* | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Instance segmentation model | Body-part segmentation model | $\text{AP}^P$ | $\text{AP}^P_{50}$ | $\text{AP}^P_{25}$ | PCP | $\text{PCP}_{50}$ | $\text{PCP}_{25}$ | $\text{AP}^H$ | $\text{AP}^H_{50}$ | $\text{AP}^H_{25}$ | $\text{mIoU}^H$ | $\text{mIoU}^P$ |
| MinkUNet [15] (Human) + Cluster | MinkUNet [15] (Body Part) | 8.9 | 34.8 | 82.5 | 8.1 | 30.6 | 58.5 | 68.2 | 83.6 | 89.4 | 92.2 | 50.8 |
| MinkUNet [15] (Body Part) + Cluster | MinkUNet [15] (Body Part) | 9.1 | 36.0 | 84.7 | 8.6 | 32.6 | 63.7 | 76.5 | 87.2 | 91.1 | 92.5 | 51.3 |
| Mask3D [65] (Human) | MinkUNet [15] (Body Part) | 5.9 | 29.9 | 90.9 | 9.2 | 33.9 | 65.4 | 95.6 | 98.7 | 99.7 | 97.6 | 53.3 |
| Mask3D [65] (Human) | KPConv [70] (Body Part) | 25.5 | 75.8 | 98.7 | 24.4 | 60.3 | 74.8 | 95.6 | 98.7 | 99.7 | 97.6 | 64.5 |
| KPConv [70] (Human) + Cluster | KPConv [70] (Body Part) | 28.2 | 74.7 | 96.3 | 22.9 | 58.4 | 73.1 | 89.7 | 95.3 | 97.0 | 96.7 | 63.6 |
| KPConv [70] (Body Part) + Cluster | KPConv [70] (Body Part) | 28.8 | 76.2 | 97.8 | 23.4 | 59.4 | 74.3 | 89.3 | 97.6 | 98.6 | 96.8 | 64.3 |
| Mask-RCNN+DeepLabv3 2D-3D (as in [86]) | | – | – | – | – | – | – | 61.3 | 97.3 | 99.8 | 87.7 | – |
| RP R-CNN 2D-3D [83] | | 26.8 | 80.5 | 97.3 | 21.8 | 61.5 | 77.6 | 74.6 | 97.2 | 97.9 | 92.1 | 58.9 |
| Human3D (Ours) | | **35.8** | **93.2** | **99.1** | **32.6** | **73.5** | **84.0** | **99.1** | **100** | **100** | **98.3** | **69.9** |

**Table 1: 3D Multi-Human Body-Part Segmentation on EgoBody test set.** Metrics are average precision for body-parts ($\text{AP}^P$) and humans ($\text{AP}^H$), correctly parsed semantic parts (PCP) and intersection-over-union on humans ($\text{IoU}^H$) and parts ($\text{IoU}^P$). Brackets indicate on which segmentation task the baselines are trained. 3D models are pre-trained on synthetic and fine-tuned on real EgoBody data.

| | *Trained on EgoBody* | | *Pre-trained on Synthetic Fine-tuned on EgoBody* | |
|---|---|---|---|---|
| Model | $\text{AP}^H$ | $\text{AP}^H_{50}$ | $\text{AP}^H$ | $\text{AP}^H_{50}$ |
| MinkUNet [15] (Human) + Cluster | 64.9 | 79.6 | 68.2 (+3.3) | 83.6 (+4.0) |
| MinkUNet [15] (Body Part) + Cluster | 69.1 | 81.7 | 76.5 (+7.4) | 87.2 (+5.5) |
| KPConv [70] (Human) + Cluster | 85.4 | 92.2 | 89.7 (+4.3) | 95.3 (+3.1) |
| KPConv [70] (Body Part) + Cluster | 86.9 | 94.4 | 89.3 (+2.4) | 97.6 (+3.2) |
| Mask3D [65] | 89.4 | **95.4** | 95.6 (+6.2) | 98.7 (+3.3) |
| Human3D (Ours) | **90.5** | 95.2 | **99.1** (+8.6) | **100** (+4.8) |

**Table 2: 3D Instance Segmentation Scores on EgoBody test.** We observe that pre-training with synthetic data results in improvements by up to +8.6 $\text{AP}^H$. Further, Human3D outperforms task-specialized models (e.g. Mask3D) by at least +3.5 $\text{AP}^H$.

| | *Trained on EgoBody* | | | *Pre-trained on Synthetic Fine-tuned on EgoBody* | | |
|---|---|---|---|---|---|---|
| Model | Scene | Human | $\text{mIoU}^H$ | Scene | Human | $\text{mIoU}^H$ |
| MinkUNet [15] | 97.5 | 85.2 | 91.3 | 98.0 | 87.9 | 92.2 (+0.9) |
| KPConv [70] | **98.9** | 93.4 | 96.1 | 99.1 | 94.4 | 96.7 (+0.6) |
| Mask3D [65] | 98.4 | 90.9 | 94.7 | 99.3 | 95.9 | 97.6 (+2.9) |
| Human3D (Ours) | 94.5 | **99.0** | **96.8** | **99.5** | **97.0** | **98.3** (+1.5) |

**Table 3: 3D Semantic Segmentation Scores on EgoBody test.** We perform binary segmentation (scene *vs.* human). We report per-class (scene *vs.* human) IoU and mean IoU ($\text{mIoU}^H$). For Mask3D and Human3D, human instance masks are merged prior to computing the semantic segmentation scores. Synthetic data pre-training results in improvements of up to +2.9 $\text{mIoU}^H$.

of prior-art methods adapted for 3D human segmentation. MinkowskiUNet [15] and KPConv [70] are voxel-based and point-based 3D semantic segmentation methods. Mask3D [65] is the state-of-the-art for 3D instance segmentation. We additionally compare with two 2D image baselines: The first one, proposed in [86], obtains human semantic masks from a pre-trained DeepLabv3 [9] applied to Kinect RGB images. Human instance masks come from a pre-trained Mask-RCNN [30]. The final 2D human instance masks are the intersection of the semantic and instance masks. Body-parts are not predicted. The second baseline, RP R-CNN [83], is a recent 2D multi-human part segmentation method. We finetune their checkpoint on our projected 2D EgoBody body-part labels. For both baselines, we backproject the 2D predictions into 3D for evaluation.

**3D Multi-Human Body-Part Segmentation (MHBPS).** Tab. 1 shows MHBPS scores of the baselines and our Human3D. The task is to detect individual human instance masks and partition them into body parts. Since there are no existing baseline models that directly predict MHBPS from point clouds, we construct strong baselines using existing 3D instance [65] and 3D semantic segmentation [15, 70] methods and solve two subtasks: Human instance masks are directly obtained from Mask3D [65] or by applying density-based clustering HDBSCAN [55, 56] on the predicted human segments (or body-part segments) from [15, 70]. MHBPS predictions are then obtained by combining human instance masks with semantic segmentation of body parts, *i.e.*, predicted body-parts inside a human mask are assigned to that human instance. Body-parts outside of any human mask are discarded.

Human3D outperforms all tested combinations of baseline methods including 2D baselines projected to 3D. Remarkably, Human3D outperforms all prior task-specific methods on 3D instance segmentation (e.g. Mask3D), and 3D semantic segmentation (e.g. KPConv) by at least +3.5 $\text{AP}^H$ and +1.6 $\text{mIoU}^H$. Human3D also significantly improves over the state-of-the-art image baseline RP R-CNN [83] that relies on RGB information and is pre-trained on much larger image datasets. Notably, we achieve these scores with depth information only. This demonstrates the benefits of Human3D operating directly on point clouds.

**3D Instance Segmentation.** Results are shown in Tab. 2. The task is to predict a set of human instances as binary foreground/background masks over the entire 3D point cloud. As before, for the 3D semantic segmentation baselines KPConv [70] and MinkUNet [15], human instances are obtained by applying density-based clustering HDBSCAN on the predicted human segments (or body-part segments) while Mask3D directly predicts human instance masks. Human3D largely outperforms all baselines tested, by at least +3.5 $\text{AP}^H$. Moreover, pre-training with synthetic data consistently improves all methods, and is particularly helpful for Human3D (+8.6 $\text{AP}^H$) which is key to improved human instance segmentation results.

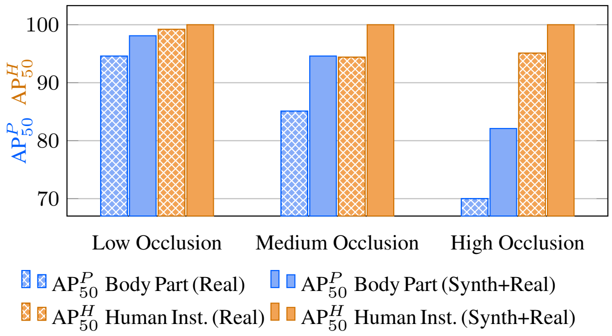


**Figure 5: Occlusion Analysis.** mAP$_{50}$ on EgoBody test on body-part segmentation 🟦 and human instance segmentation 🟧 for Human3D with and without pre-training on synthetic data. Pre-training on synthetic data is particularly helpful for highly occluded humans, *e.g.*, part segmentation improves by +12.1 $\text{AP}^P_{50}$.

| | | *3D Instance Segmentation* | | *3D Semantic Segmentation* |
|---|---|---|---|---|
| Pre-Training Data | Fine-Tuning Data | $\text{AP}^H$ | $\text{AP}^H_{50}$ | $\text{mIoU}^H$ |
| ① – | Real (EgoBody) | 89.4 | 95.4 | 94.7 |
| ② Real (BEHAVE) | Real (EgoBody) | 92.0 | 96.8 | 96.8 |
| ③ Real (EgoBody) | Real (EgoBody) | 91.8 | 96.9 | 95.8 |
| ④ Synthetic (ours) | Real (EgoBody) | **95.6** | **98.7** | **97.6** |

**Table 4: Training Settings Analysis.** We compare pre-training on synthetic and real data for instance and semantic segmentation.

**3D Semantic Segmentation.** Tab. 3 shows binary (scene *vs.* human) segmentation results with and without pre-training on synthetic data. We adapt Mask3D [65] and Human3D by merging predicted human instance masks with confidence scores above 50% before computing semantic segmentation scores. We observe that Human3D significantly outperforms specialized semantic segmentation models [15,70] by at least +1.6 $\text{mIoU}^H$. Intuitively, Human3D has the potential to leverage the body-part annotations as an additional supervision signal. Again, we find that pre-training with synthetic data enhances the performance of all models.

## 5.2. Analysis Experiments

**Does synthetic data help with occlusions?** Occlusions are a main challenge in cluttered indoor spaces. In Fig. 5, we analyze the influence of synthetic training data on occluded humans. One key advantage of synthetic data is that it can be tailored to specific edge cases that are rare in real data. Our synthetic data contains numerous people in real cluttered scenes and therefore numerous occlusions. To evaluate the effect of occlusions, we further split our test data into three groups of increasing levels of human occlusions: *low* (122 scenes), *medium* (104 scenes), *high* (78 scenes). Details are in the supplementary. Pre-training with synthetic data drastically improves body-part segmentation (+12.1 $\text{AP}^P_{50}$) and human instance segmentation (+4.9 $\text{AP}^H_{50}$) performance for highly occluded humans.

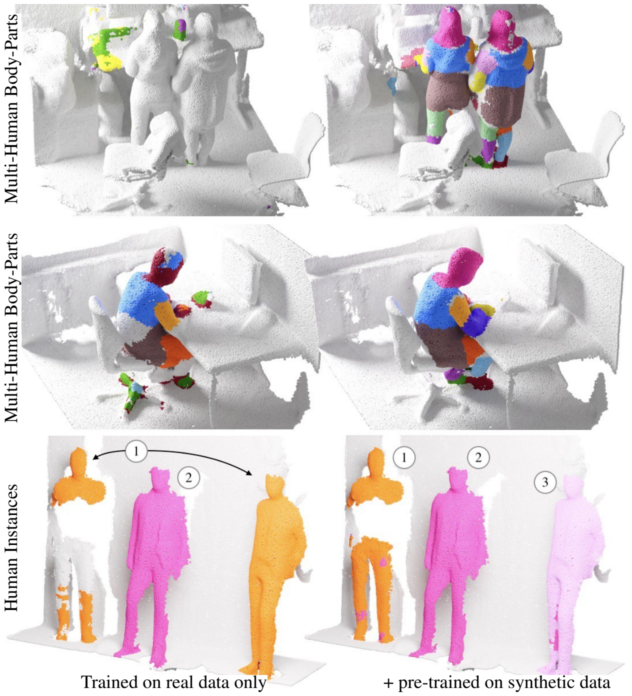


**Figure 6: Effect of Synthetic Data.** Model trained only on real EgoBody data *(left)* and additionally pre-trained on synthetic data *(right)*. Synthetic pre-training improves robustness for close interactions of humans *(top)* or human-scene interactions *(middle)*, and improves generalization to multiple people *(bottom)*.

**Does synthetic data improve generalization?** To keep labeling efforts within limits, EgoBody [86] does not contain humans that are too closely interacting with other humans or objects, and is limited to two humans per scene. A key question is whether synthetic data can help to generalize beyond these limitations of the real-world training scenes. Fig. 6 depicts these edge cases and shows improved performance when comparing our Human3D with and without pre-training on synthetic data. The pre-trained model is able to segment humans that are closely interacting *(top)*, a person that is in close contact with a desk and thus heavily occluded *(middle)*, and can successfully segment more than two people where the model trained on real data assigns the same instance label to two different people *(bottom)*.

**Pre-training on synthetic or real data?** In a preliminary study (Tab. 4), we compare different settings for pre-training on 3D instance and semantic segmentation using [65]. We always fine-tune on the real EgoBody training set. The baseline ① does not include any pre-training. Model ④ pre-trained on synthetic data provides the biggest boost over ① (+6.2 $\text{AP}^H$, +2.9 mIoU). To verify that the improvement is not due to more training iterations or better weight initialization, we repeat the experiment and use EgoBody also for pre-training ③ as well as another real dataset BEHAVE ②. We see that ② and ③ perform comparably. Importantly, however, pre-training on synthetic data ④ improves signif-

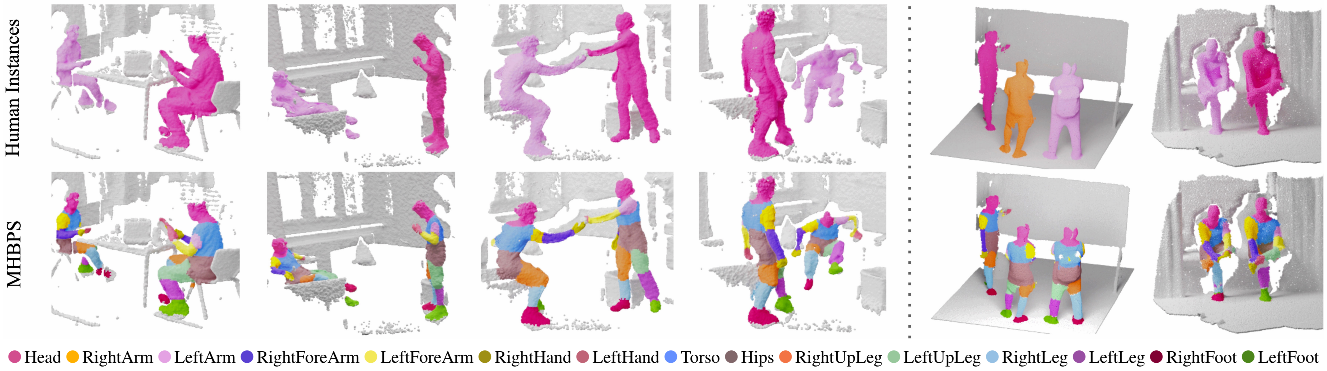

Human Instances
MHBPS
● Head ● RightArm ● LeftArm ● RightForeArm ● LeftForeArm ● RightHand ● LeftHand ● Torso ● Hips ● RightUpLeg ● LeftUpLeg ● RightLeg ● LeftLeg ● RightFoot ● LeftFoot

**Figure 7: Human3D Qualitative Results.** Human instance segmentation results *(top)* and multi-human body-part segmentation results *(bottom)* on point clouds from Kinect sensors from our EgoBody test set *(left)* and on out-of-domain point clouds from iPhone LiDAR scans *(right)*. The rightmost example shows a failure case where the left and right legs are confused due to the person crossing their legs.

| Two-stage | Restricted | *Multi-Human Body-Part Seg.* | | | |
|---|---|---|---|---|---|
| Hungarian Matching | Cross-Attention | $\text{AP}^P$ | $\text{AP}^P_{50}$ | PCP | $\text{PCP}_{50}$ |
| ✓ (two-stage) | ✓ | 33.7 | **82.3** | 30.8 | 66.9 |
| ✓ (two-stage) | ✗ | **34.0** | 79.5 | **31.1** | **78.1** |
| ✗ (one-stage) | ✗ | 2.0 | 12.5 | 2.2 | 8.0 |

**Table 5: Human3D Ablation Study.** Hungarian matching and attention mechanisms. Models trained on EgoBody, no pre-training.

icantly over pre-training on EgoBody ③ and BEHAVE ②, proving the importance of synthetic pre-training.

**Human3D Ablation Study.** In Tab. 5, we analyze design choices of Human3D, *i.e.*, the masked attention module, and Hungarian matching. The study reveals that our newly proposed *two-stage* Hungarian matching is crucial for MHBPS. When using the existing *single*-stage Hungarian matching (as in [7, 65]), body-part queries and human queries of the same human can be falsely assigned to two different ground truth humans. Instead, our two-stage Hungarian matching guarantees consistent supervision such that human queries and the corresponding body-part queries are always supervised by a single ground truth human. The effect of restricting the cross-attention between body-part queries and point features to lie within the corresponding human mask is less significant but improves $\text{AP}^P_{50}$ scores.

### 5.3. Qualitative Results and Discussion

Fig. 7 shows qualitative results of Human3D for 3D instance segmentation and 3D multi-human body-part segmentation. Our model works on point clouds from Kinect depth sensors *(left)* and generalizes to out-of-domain point clouds as shown by the scans from the iPhone LiDAR sensor *(right)*. Human3D is able to clearly segment closely interacting humans, under strong occlusions, and in close contact with scene objects such as sofas or chairs. This is also reflected in the scores reported in Tab. 1. The body-part segmentation can fail when people cross their legs (*i.e.*, left/right confusion). Additional qualitative results are provided in the supplementary material Sec. 5.

**Limitations.** Our unified Human3D shows considerable improvements over combinations of specialized state-of-the-art 3D segmentation methods; however, several limitations remain. Our method focuses on segmenting humans and body parts, while other works [15, 65, 70] primarily focus on 3D scene segmentation. In this context, it would be interesting to explore a unified approach that jointly predicts segmentation for both humans and scenes. Similar to existing work for placing humans into 3D scenes [29, 77, 87], our pipeline generates humans with minimal clothing. To obtain more realistic training data, a promising avenue would be to integrate the generation of clothed humans [10, 53].

## 6 Conclusion

In this work, we have introduced Human3D, the first unified model for end-to-end 3D multi-human body-part segmentation, operating directly on point clouds. The key novelties of our transformer-based model are the two-level queries representing human and body-part instances, as well as the two-stage Hungarian matching for supervision. Using our synthetic training data generation framework, we have further shown that pre-training on synthetic training data can significantly improve 3D human segmentation performance on various tasks and models, especially in challenging conditions such as strong occlusion. We believe that Human3D is an important step towards holistic 3D scene understanding with human-scene interactions.

**Acknowledgments.** This project is funded by Innosuisse (48727.1 IP-ICT), ERC CoG grant DeeViSe (ERC-2017-CoG-773161), BMBF project 6GEM (16KISK036K), SNF Grant 200021 204840, and compute resources from RWTH Aachen (rwth1261). We sincerely thank Siwei Zhang for helping with the EgoBody dataset, Anne Marx and Theodora Kontogianni for providing guidance on the 3D annotation tool, and István Sárándi for helpful discussions. Francis Engelmann is a post-doctoral research fellow at the ETH AI Center.

## 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. 2, 3, 14

[2] Matan Atzmon, Haggai Maron, and Yaron Lipman. Point Convolutional Neural Networks by Extension Operators. In *ACM Transactions On Graphics (TOG)*, 2018. 2

[3] Eduard Gabriel Bazavan, Andrei Zanfir, Mihai Zanfir, William T. Freeman, Rahul Sukthankar, and Cristian Sminchisescu. HSPACE: Synthetic Parametric Humans Animated in Complex Environments. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[4] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *International Conference on Computer Vision (ICCV)*, 2019. 2

[5] 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, 15, 17

[6] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A Multimodal Dataset for Autonomous Driving. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-End Object Detection with Transformers. In *European Conference on Computer Vision (ECCV)*, 2020. 5, 6, 9

[8] 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. 2, 3, 14

[9] 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. 7, 20

[10] Xu Chen, Tianjian Jiang, Jie Song, Jinlong Yang, Michael J. Black, Andreas Geiger, and Otmar Hilliges. gDNA: Towards Generative Detailed Neural Avatars. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 9

[11] Xiaojia Chen, Xuanhan Wang, Lianli Gao, and Jingkuan Song. RepParser: End-to-End Multiple Human Parsing with Representative Parts. *arXiv preprint arXiv:2208.12908*, 2022. 1, 2

[12] 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, 19

[13] Julian Chibane, Francis Engelmann, Tuan Anh Tran, and Gerard Pons-Moll. Box2Mask: Weakly Supervised 3D Semantic Instance Segmentation using Bounding Boxes. In *European Conference on Computer Vision (ECCV)*, 2022. 2

[14] Benjamin Choi, Çetin Meriçli, Joydeep Biswas, and Manuela Veloso. Fast Human Detection for Indoor Mobile Robots Using Depth Images. In *International Conference on Robotics and Automation (ICRA)*, 2013. 1, 2

[15] 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. 2, 5, 7, 8, 9, 19

[16] 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. 2, 3, 6, 14, 15

[17] Ruoxi Deng, Chunhua Shen, Shengjun Liu, Huibing Wang, and Xinru Liu. Learning to Predict Crisp Boundaries. In *European Conference on Computer Vision (ECCV)*, 2018. 6

[18] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An Open Urban Driving Simulator. In *Conference on Robot Learning*, 2017. 3

[19] Cathrin Elich, Francis Engelmann, Theodora Kontogianni, and Bastian Leibe. 3D-BEVIS: Birds-Eye-View Instance Segmentation. In *German Conference on Pattern Recognition (GCPR)*, 2019. 2

[20] Francis Engelmann, Martin Bokeloh, Alireza Fathi, Bastian Leibe, and Matthias Nießner. 3D-MPA: Multi Proposal Aggregation for 3D Semantic Instance Segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[21] Francis Engelmann, Theodora Kontogianni, Alexander Hermans, and Bastian Leibe. Exploring Spatial Context for 3D Semantic Segmentation of Point Clouds. In *International Conference on Computer Vision (ICCV) Workshops*, 2017. 2

[22] Francis Engelmann, Theodora Kontogianni, Jonas Schult, and Bastian Leibe. Know What Your Neighbors Do: 3D Semantic Segmentation of Point Clouds. In *European Conference on Computer Vision (ECCV) Workshops*, 2018. 2

[23] Ke Gong, Xiaodan Liang, Yicheng Li, Yimin Chen, Ming Yang, and Liang Lin. Instance-level Human Parsing via Part Grouping Network. In *European Conference on Computer Vision (ECCV)*, 2018. 1, 2

[24] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[25] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. DensePose: Dense Human Pose Estimation in the Wild. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[26] Lei Han, Tian Zheng, Lan Xu, and Lu Fang. OccuSeg: Occupancy-aware 3D Instance Segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[27] 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. 15

[28] 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. 3, 4, 15

[29] Mohamed Hassan, Partha Ghosh, Joachim Tesch, Dimitrios Tzionas, and Michael J. Black. Populating 3D Scenes by Learning Human-Scene Interaction. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3, 9

[30] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *International Conference on Computer Vision (ICCV)*, 2017. 1, 2, 7, 20

[31] David T. Hoffmann, Dimitrios Tzionas, Michael J. Black, and Siyu Tang. Learning to Train with Synthetic Humans. In *German Conference on Pattern Recognition (GCPR)*, 2019. 3

[32] Ji Hou, Angela Dai, and Matthias Nießner. 3D-SIS: 3D Semantic Instance Segmentation of RGB-D Scans. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[33] Zeyu Hu, Xuyang Bai, Jiaxiang Shang, Runze Zhang, Jiayu Dong, Xin Wang, Guangyuan Sun, Hongbo Fu, and Chiew Lan Tai. VMNet: Voxel-Mesh Network for Geodesic-Aware 3D Semantic Segmentation. In *International Conference on Computer Vision (ICCV)*, 2021. 2

[34] Binh-Son Hua, Minh-Khoi Tran, and Sai-Kit Yeung. Pointwise Convolutional Neural Network. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[35] Chun-Hao P. Huang, Hongwei Yi, Markus Höschle, Matvey Safroshkin, Tsvetelina Alexiadis, Senya Polikovsky, Daniel Scharstein, and Michael J. Black. Capturing and Inferring Dense Full-Body Human-Scene Contact. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[36] Jing Huang and Suya You. Point Cloud Labeling Using 3D Convolutional Neural Network. In *International Conference on Pattern Recognition (ICPR)*, 2016. 2

[37] Andrew Hynes and Stephen Czarnuch. Human Part Segmentation in Depth Images with Annotated Part Positions. In *Sensors*, 2018. 1, 2

[38] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. PointGroup: Dual-Set Point Grouping for 3D Instance Segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[39] Shaharyar Kamal, Ahmad Jalal, and Cesar Azurdia-Meza. Depth Maps-Based Human Segmentation and Action Recognition Using Full-Body Plus Body Color Cues via Recognizer Engine. In *Journal of Electrical Engineering and Technology*, 2019. 1, 2

[40] 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, 6, 18

[41] Hema Koppula, Abhishek Anand, Thorsten Joachims, and Ashutosh Saxena. Semantic Labeling of 3D Point Clouds for Indoor Scenes. In *Neural Information Processing Systems (NeurIPS)*, 2011. 2

[42] Lars Kreuzberg, Idil Esen Zulfikar, Sabarinath Mahadevan, Francis Engelmann, and Bastian Leibe. 4D-StOP: Panoptic Segmentation of 4D LiDAR using Spatio-temporal Object Proposal Generation and Aggregation. *European Conference on Computer Vision (ECCV)*, 2022. 2

[43] Harold W. Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955. 5, 6

[44] Jean Lahoud, Bernard Ghanem, Marc Pollefeys, and Martin R. Oswald. 3D Instance Segmentation via Multi-task Metric Learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[45] Loic Landrieu and Martin Simonovsky. Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[46] Jianshu Li, Jian Zhao, Yunchao Wei, Congyan Lang, Yidong Li, Terence Sim, Shuicheng Yan, and Jiashi Feng. Multiple-Human Parsing in the Wild. *arXiv preprint arXiv:1705.07206*, 2017. 6

[47] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. PointCNN: Convolution on X-transformed Points. In *Neural Information Processing Systems (NeurIPS)*, 2018. 2

[48] Liqiang Lin, Yilin Liu, Yue Hu, Xingguang Yan, Ke Xie, and Hui Huang. Capturing, Reconstructing, and Simulating: the UrbanScene3D Dataset. In *European Conference on Computer Vision (ECCV)*, 2022. 3

[49] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[50] 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. 2, 4, 17

[51] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations (ICLR)*, 2019. 6

[52] Yan Lu and Christopher Rasmussen. Simplified Markov Random Fields for Efficient Semantic Labeling of 3D Point Clouds. In *International Conference on Intelligent Robots and Systems (ICIRS)*, 2012. 2

[53] Qianli Ma, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael J. Black. Learning to Dress 3D People in Generative Clothing. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 9

[54] Naureen Mahmood, Yehonal Azeroth, Sricharan Chiruvolu, and Denis Heid. Meshcapade Wiki. https://github.com/Meshcapade/wiki. Accessed: 2022-11-10. 4, 16, 17

[55] Leland McInnes and John Healy. Accelerated Hierarchical Density Based Clustering. In *International Conference on Data Mining Workshops (ICDMW)*, 2017. 7, 19

[56] Leland McInnes, John Healy, and Steve Astels. HDBSCAN: Hierarchical Density Based Clustering. *The Journal of Open Source Software*, 2017. 7, 19

[57] 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. 19

[58] Alexey Nekrasov, Jonas Schult, Or Litany, Bastian Leibe, and Francis Engelmann. Mix3D: Out-of-Context Data Augmentation for 3D Scenes. In *International Conference on 3D Vision (3DV)*, 2021. 2

[59] 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 *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[60] 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, 16, 17

[61] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[62] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Neural Information Processing Systems (NeurIPS)*, 2017. 2

[63] Anurag Ranjan, David T. Hoffmann, Dimitrios Tzionas, Siyu Tang, Javier Romero, and Michael J. Black. Learning Multi-Human Optical Flow. In *International Journal of Computer Vision (IJCV)*, 2020. 3

[64] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015. 6

[65] 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. 2, 5, 7, 8, 9, 19, 20

[66] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time Human Pose Recognition in Parts from Single Depth Images. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. 1, 2

[67] Leslie N. Smith and Nicholay Topin. Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, 2019. 6

[68] Lyne P. Tchapmi, Christopher B. Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. SEGCloud: Semantic Segmentation of 3D Point Clouds. In *International Conference on 3D Vision (3DV)*, 2017. 2

[69] Purva Tendulkar, Dídac Surís, and Carl Vondrick. FLEX: Full-Body Grasping Without Full-Body Grasps. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3

[70] 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. 2, 7, 8, 9, 19

[71] Denis Tome, Patrick Peluse, Lourdes Agapito, and Hernan Badino. xR-EgoPose: Egocentric 3D Human Pose from an HMD Camera. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[72] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from Synthetic Humans. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[73] Thang Vu, Kookhoi Kim, Tung M. Luu, Xuan Thanh Nguyen, and Chang D. Yoo. SoftGroup for 3D Instance Segmentation on 3D Point Clouds. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[74] 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. 3, 14

[75] Tianyi Wang, Jian Li, and Xiangjing An. An Efficient Scene Semantic Labeling Approach for 3D Point Cloud. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2015. 2

[76] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. SGPN: Similarity Group Proposal Network for 3D Point Cloud Instance Segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[77] Zan Wang, Yixin Chen, Tengyu Liu, Yixin Zhu, Wei Liang, and Siyuan Huang. HUMANISE: Language-conditioned Human Motion Generation in 3D Scenes. In *Neural Information Processing Systems (NeurIPS)*, 2022. 3, 9

[78] Daniel Wolf, Johann Prankl, and Markus Vincze. Fast Semantic Segmentation of 3D Point Clouds using a Dense CRF with Learned Parameters. In *International Conference on Robotics and Automation (ICRA)*, 2015. 2

[79] Erroll Wood, Tadas Baltrušaitis, Charlie Hewitt, Sebastian Dziadzio, Matthew Johnson, Virginia Estellers, Thomas J. Cashman, and Jamie Shotton. Fake It Till You Make It: Face Analysis in the Wild Using Synthetic Data Alone. In *International Conference on Computer Vision (ICCV)*, 2021. 2, 3

[80] Lu Xia, Chia-Chih Chen, and J. K. Aggarwal. Human Detection Using Depth Information by Kinect. In *Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2011. 1, 2

[81] Yifan Xu, Tianqi Fan, Mingye Xu, Long Zeng, and Yu Qiao. SpiderCNN: Deep Learning on Point Sets with Parameterized Convolutional Filters. In *European Conference on Computer Vision (ECCV)*, 2018. 2

[82] Bo Yang, Jianan Wang, Ronald Clark, Qingyong Hu, Sen Wang, Andrew Markham, and Niki Trigoni. Learning Object Bounding Boxes for 3D Instance Segmentation on

Point Clouds. In *Neural Information Processing Systems (NeurIPS)*, 2019. 2

[83] Lu Yang, Qing Song, Zhihui Wang, Mengjie Hu, Chun Liu, Xueshi Xin, Wenhe Jia, and Songcen Xu. Renovating Parsing R-CNN for Accurate Multiple Human Parsing. In *European Conference on Computer Vision (ECCV)*, 2020. 1, 2, 7

[84] Lu Yang, Qing Song, Zhihui Wang, and Ming Jiang. Parsing R-CNN for Instance-Level Human Analysis. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2

[85] Yuanwen Yue, Sabarinath Mahadevan, Jonas Schult, Francis Engelmann, Bastian Leibe, Konrad Schindler, and Theodora Kontogianni. AGILE3D: Attention Guided Interactive Multi-object 3D Segmentation. *arXiv preprint arXiv:2306.00977*, 2023. 2

[86] 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, 6, 7, 8, 15, 17, 18, 20

[87] 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. 3, 9, 14, 15

[88] Yan Zhang, Mohamed Hassan, Heiko Neumann, Michael J. Black, and Siyu Tang. Generating 3D People in Scenes without People. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[89] Jian Zhao, Jianshu Li, Yu Cheng, Terence Sim, Shuicheng Yan, and Jiashi Feng. Understanding Humans in Crowded Scenes: Deep Nested Adversarial Learning and A New Benchmark for Multi-Human Parsing. In *ACM International Conference on Multimedia*, 2018. 1, 2

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

*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. We will release our code, model and data for research purposes.*

## 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 [16], which is a large-scale 3D indoor dataset. The ScanNet [16] 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 [16] 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, 8, 74], 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 [87], 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 [16]. 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.

**PLACE: Proximity Learning of Articulation and Contact in 3D Environments [87].** For placing synthetic humans, we leverage PLACE [87], which is a generative human-scene interaction synthesis method. Given a 3D scene without humans, generation and placement of synthetic humans using PLACE [87] 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 [87], 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 [87] for details.

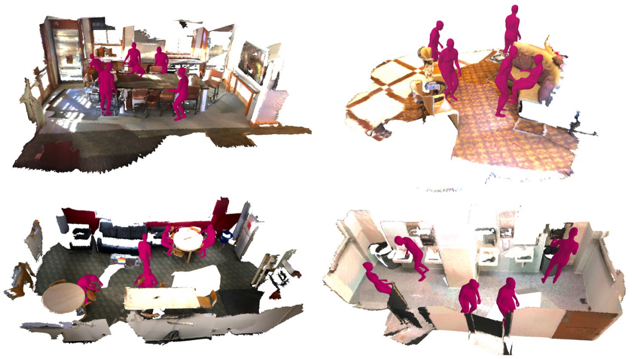


**Figure 1: Synthetic Humans in ScanNet [16] Scenes.** Example scenes (dining room, kitchen, living room, bathroom) populated with synthetic humans using PLACE [87] with instance-segmentation guided human location sampling.

**Modified PLACE: Instance Segmentation Guided Human Location Sampling.** PLACE [87] 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 [87] 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 [16] 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 $n_{\text{objects}}$, 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 $n_{\text{humans}}$ objects. Otherwise, we select all available ($n_{\text{objects}}$) 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 [87].

(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

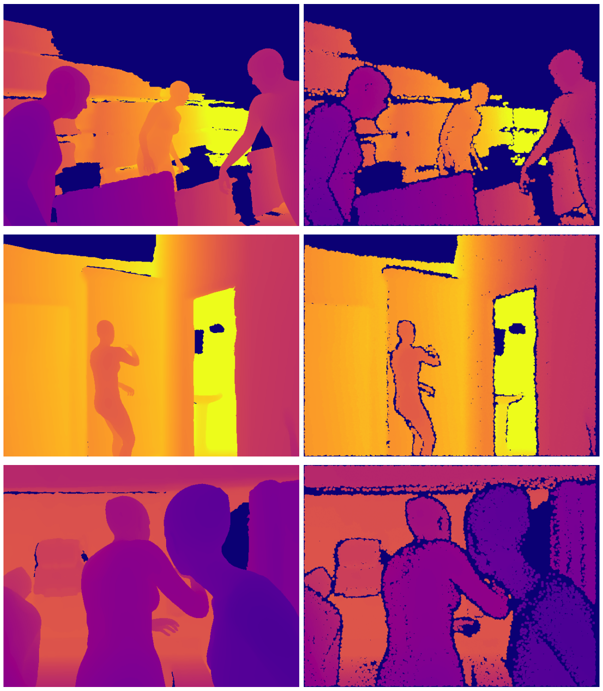
We are primarily interested in creating a labeled synthetic dataset of partial point clouds obtained from depth scans. In order to obtain realistic depth maps and corresponding label images, we need to place a virtual camera in each scene with synthetic humans, and render frames using this virtual camera. 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 $480 \times 640$ ($h \times w$) 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 [28] – particularly the implementation available at [27]. 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.



(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.

## 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 [16] 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 [86] only has 2 humans per scene, and the BEHAVE [5] 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.

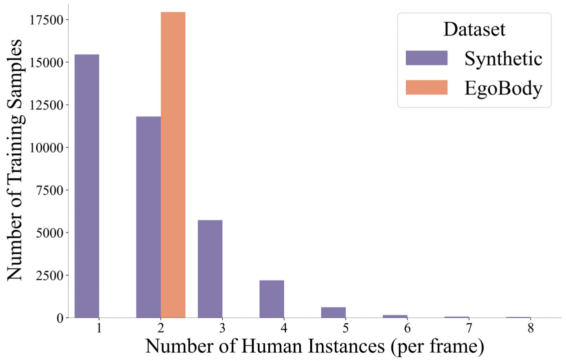


**Figure 3: Number of human instances vs. number of training samples.** Our synthetic dataset features scenes with up to 8 human instances whereas each EgoBody scene features exactly 2 subjects.

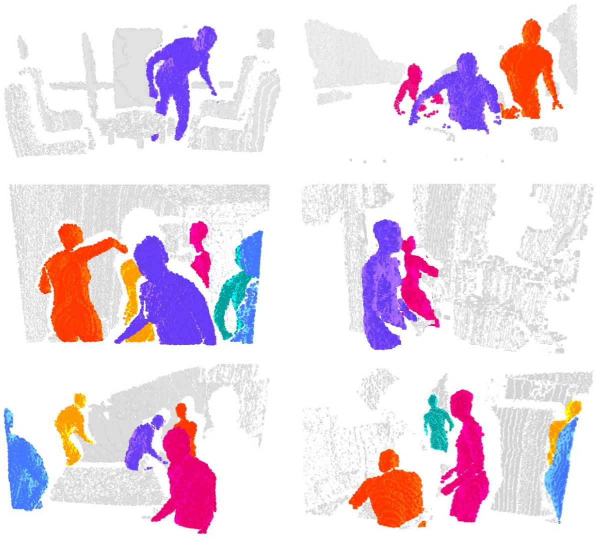


**Figure 4: Example synthetic training scenes (point clouds).** Our synthetic dataset features point clouds with a varying number of human instances.

## 1.5 Merging Body Parts

In order to obtain body-part labels, we first map the faces of each SMPL-X [60] mesh to 26 body parts according to the mapping in [54]. Afterwards, we merge smaller body parts into larger ones as shown in Tab. 1 and Fig. 5, and 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)

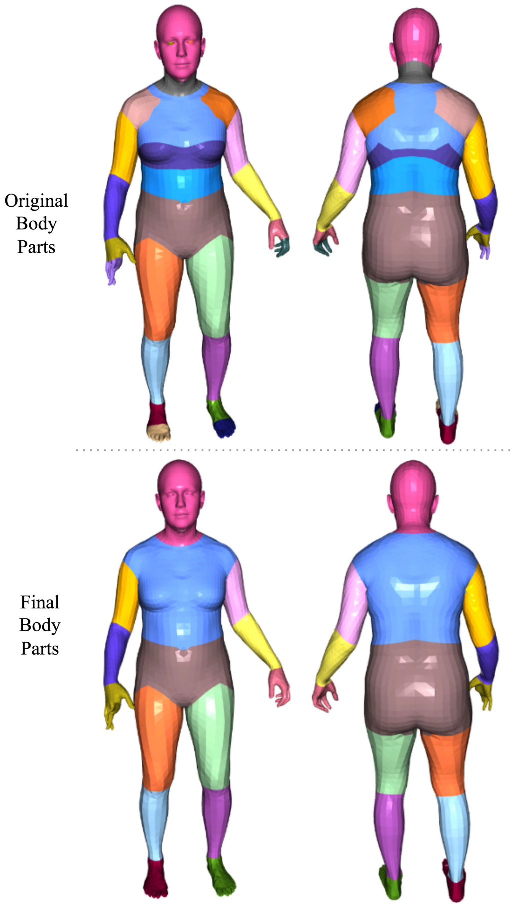


**Figure 5: Illustration of body part merging.** The first row shows original body parts, and the second row shows the body parts obtained after merging smaller parts into larger ones.

## 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 two labels: background and human. For multi-human body-part segmentation, we map the faces of each SMPL-X [60] mesh to body-parts according to the mapping described in Sec. 1.5 and assign each point to one of the 15 body parts.

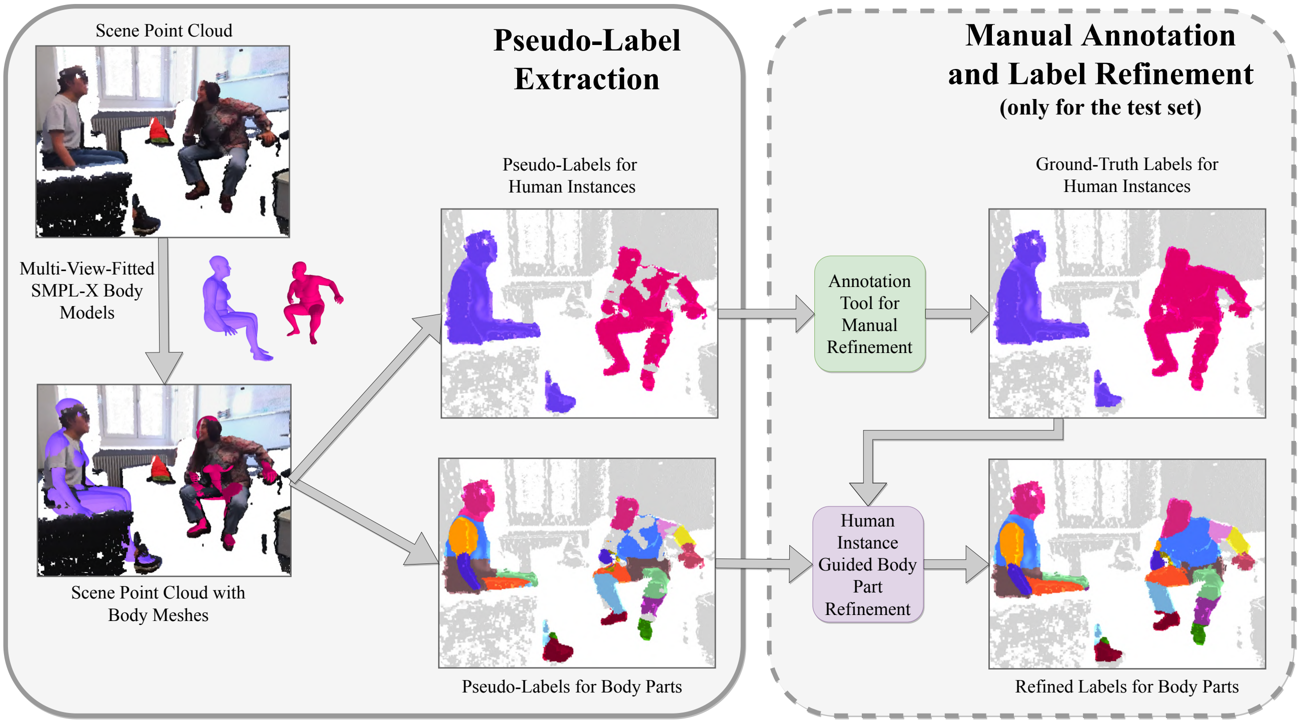

Pseudo-Label Extraction
Scene Point Cloud
Multi-View-Fitted SMPL-X Body Models
Scene Point Cloud with Body Meshes
Pseudo-Labels for Human Instances
Pseudo-Labels for Body Parts
Manual Annotation and Label Refinement (only for the test set)
Annotation Tool for Manual Refinement
Ground-Truth Labels for Human Instances
Human Instance Guided Body Part Refinement
Refined Labels for Body Parts

**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.

### 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.

### 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 [86] and BEHAVE [5] datasets. Our pipeline for extracting pseudo-labels is illustrated in Fig. 6 (left block).

**EgoBody [86].** 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 [60] 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 [60] body mesh to body parts according to the mapping in [54], 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 [5].** 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 [50] 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 [5] 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 [50] mesh to body parts according to the mapping in [54], 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.

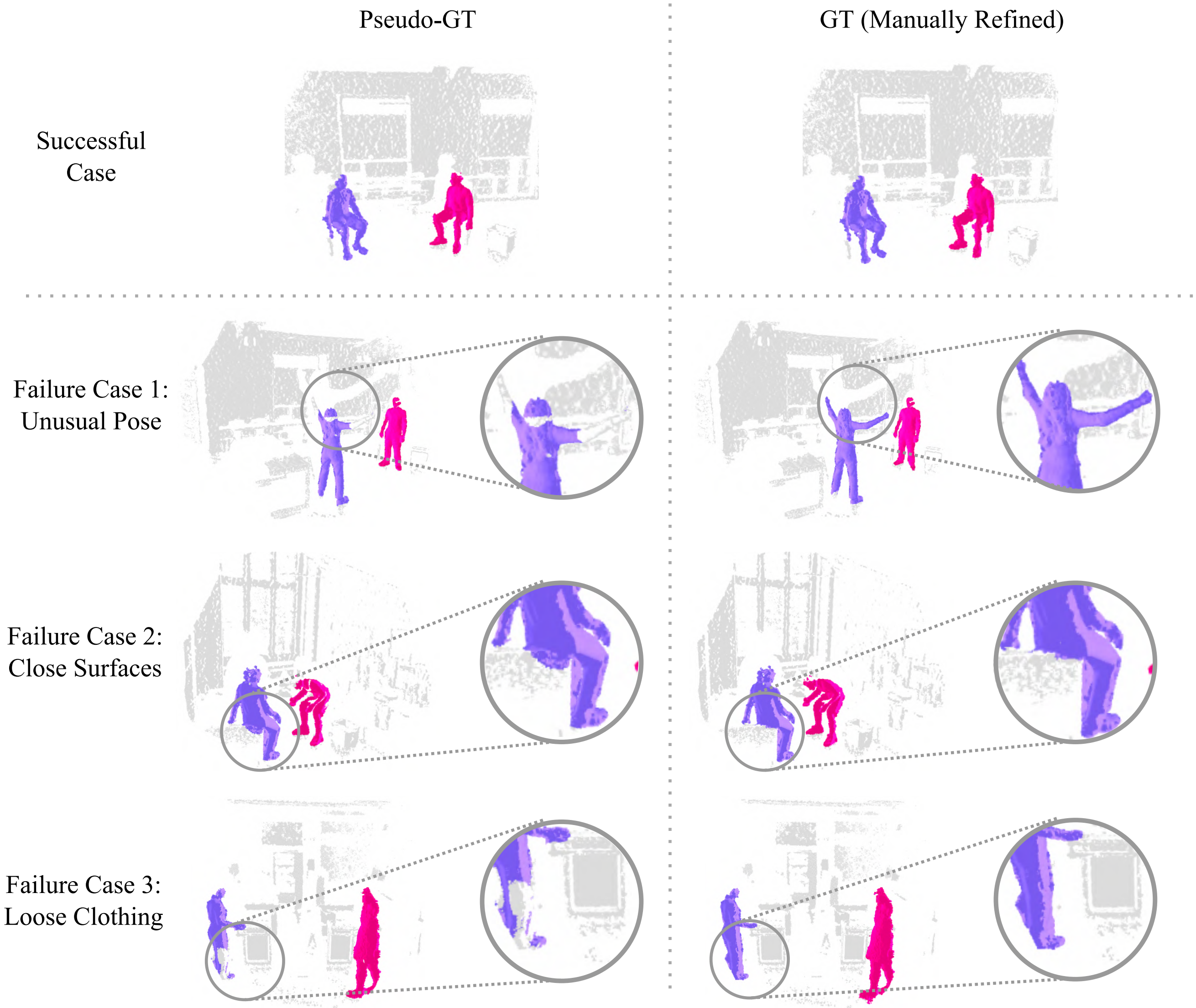

Pseudo-GT
GT (Manually Refined)
Successful Case
Failure Case 1: Unusual Pose
Failure Case 2: Close Surfaces
Failure Case 3: Loose Clothing

**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.

### 2.2 Manually Refined Evaluation Dataset

The EgoBody [86] 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 [40]. 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 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 $\text{AP}^H$ : 91.9, $\text{AP}^H_{50}$ : 99.3, and $\text{AP}^H_{25}$ : 99.5, highlighting the need for manual annotations.

## 3 Human3D Architecture Details

We obtain strong multi-scale point features from a Minkowski Res16UNet18B [15]. 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 [65], we also use the modified transformer decoder of Mask2Former [12] instantiated with 8-headed attention and a feedforward network of 1024 dimensions. We sample point features for the cross-attention following Mask3D [65]. 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 [57, 65], 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 [70] and MinkUNet [15], we obtain human instances by applying density-based clustering HDBSCAN [55,56] 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 pre-training with synthetic data with respect to different human 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 ($\text{AP}^P_{50}$) 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,<br>stand up | lean,<br>lie down | dance,<br>exercise | kneel,<br>bend | pick, put,<br>hold an object | reach, touch,<br>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 ($\text{AP}^P_{50}$) with and without synthetic pre-training. We observe the largest improvement for actions that cause significant self-occlusions such as walking and bending.

### 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

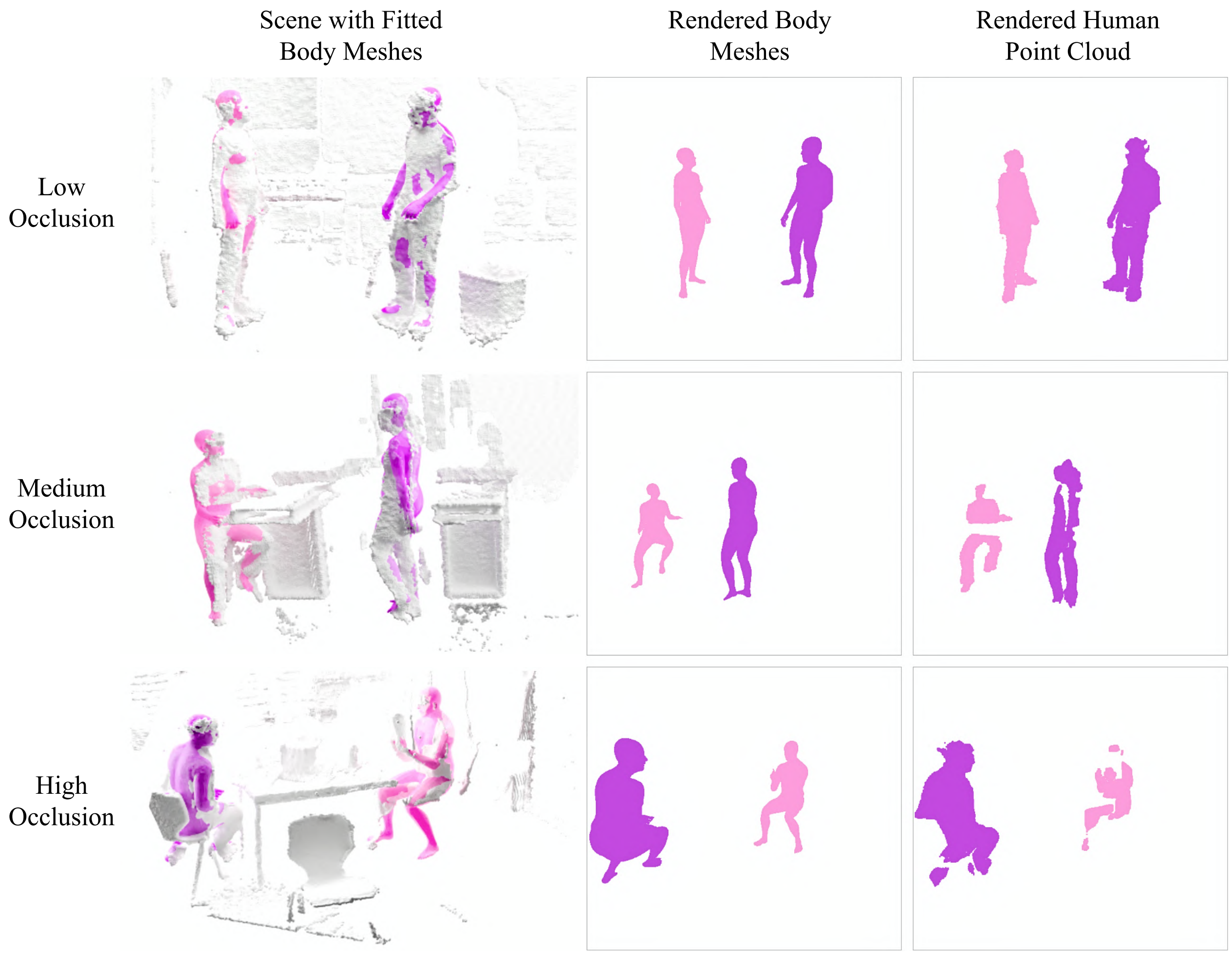


**Figure 8: Occlusion Computation.** In the first column, SMPL-X human body meshes fitted to humans in the EgoBody dataset are shown. The fitted body meshes for each human (*second* column) as well as human masks (*third* column) obtained from our manual annotation of the scene point clouds are projected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

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.

In this section, we provide further implementation details about the *Mask-RCNN+DeepLabv3 2D-3D* baseline. This baseline closely follows the approach from [86]. The human semantic segmentation is obtained by applying a pretrained DeepLabv3 [9] to the Kinect RGB image. To obtain human instances, a pretrained Mask-RCNN is applied [30]. 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 [65] 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

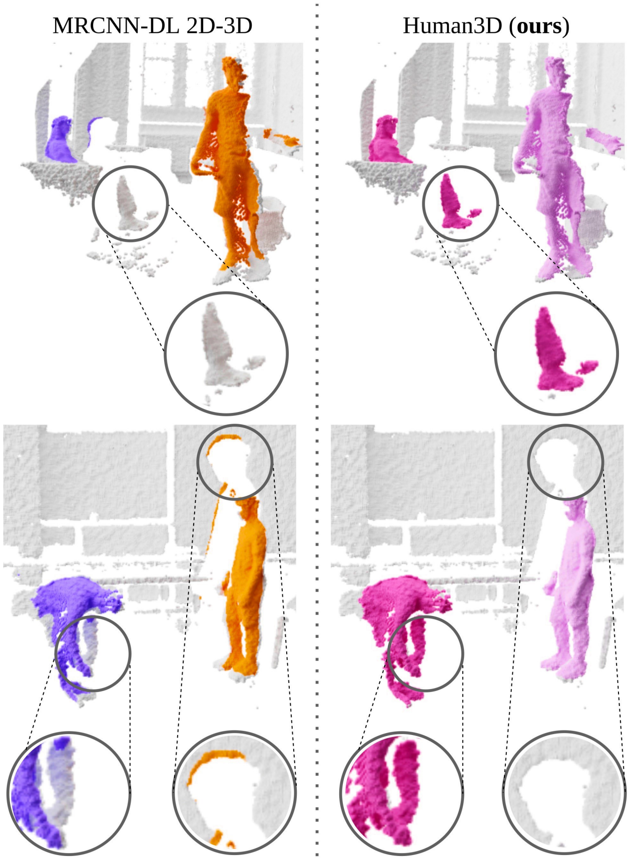


**Figure 9: Failure cases of the 2D baseline.** The first row shows a typical error of the Mask-RCNN baseline. The sofa occludes most of the human resulting in an incomplete human mask. In addition, the second example shows that small errors at the boundaries in 2D lead to incorrectly predicted 3D points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

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 | $\mathrm{AP}^H$ | $\mathrm{AP}^H_{50}$ |
|---|---|---|---|
| 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.

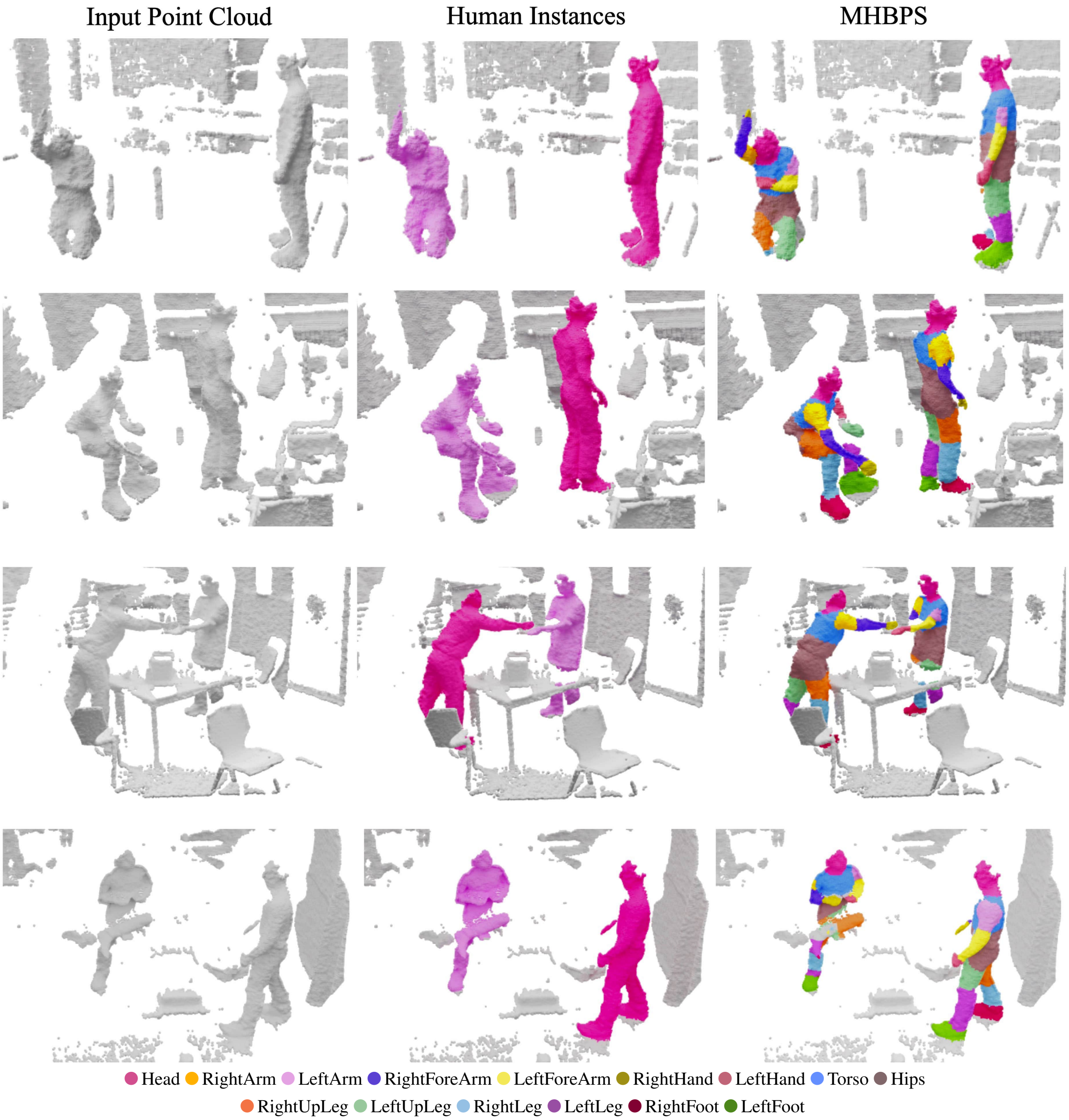

Input Point Cloud
Human Instances
MHBPS
● Head ● RightArm ● LeftArm ● RightForeArm ● LeftForeArm ● RightHand ● LeftHand ● Torso ● Hips
● RightUpLeg ● LeftUpLeg ● RightLeg ● LeftLeg ● RightFoot ● LeftFoot

**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.

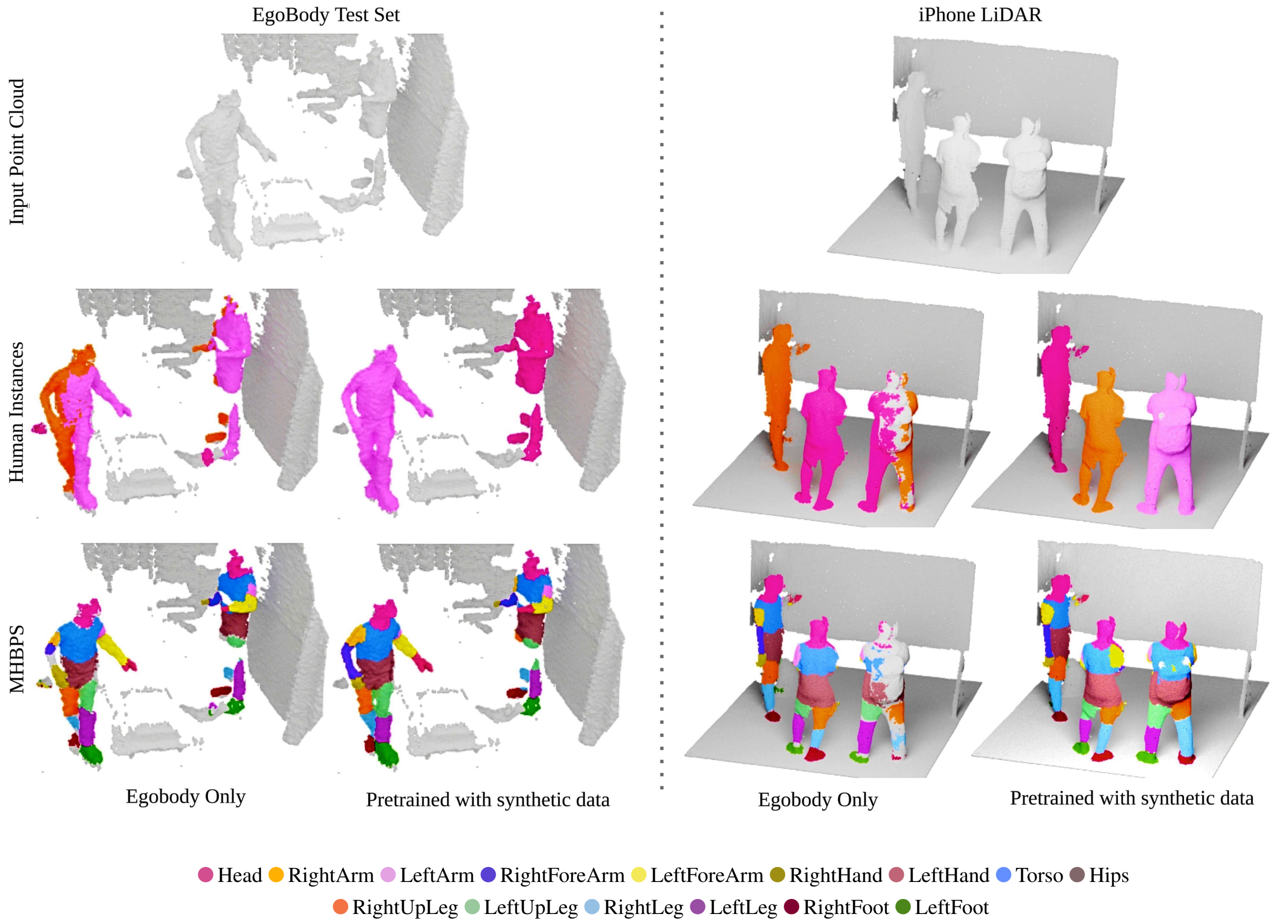

EgoBody Test Set
iPhone LiDAR
Input Point Cloud
Human Instances
MHBPS
Egobody Only
Pretrained with synthetic data
Egobody Only
Pretrained with synthetic data
● Head ● RightArm ● LeftArm ● RightForeArm ● LeftForeArm ● RightHand ● LeftHand ● Torso ● Hips
● RightUpLeg ● LeftUpLeg ● RightLeg ● LeftLeg ● RightFoot ● LeftFoot

**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.

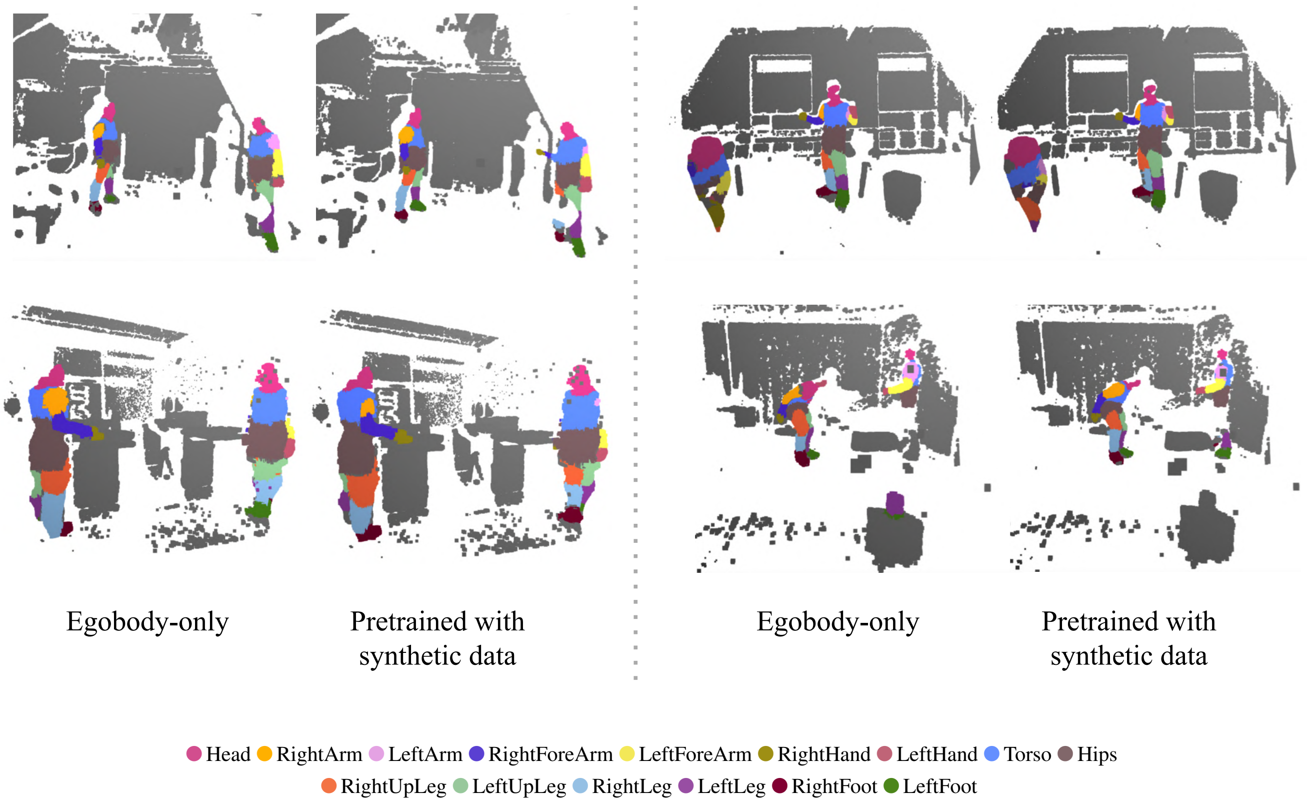

Egobody-only
Pretrained with synthetic data
Egobody-only
Pretrained with synthetic data
● Head ● RightArm ● LeftArm ● RightForeArm ● LeftForeArm ● RightHand ● LeftHand ● Torso ● Hips
● RightUpLeg ● LeftUpLeg ● RightLeg ● LeftLeg ● RightFoot ● LeftFoot

**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.