Learning to Segment Humans in 3D Scenes

Cafer Mertcan Akçay
cakcay@ethz.ch

İrem Kaftan
ikaftan@ethz.ch

Ayça Takmaz
takmaza@ethz.ch

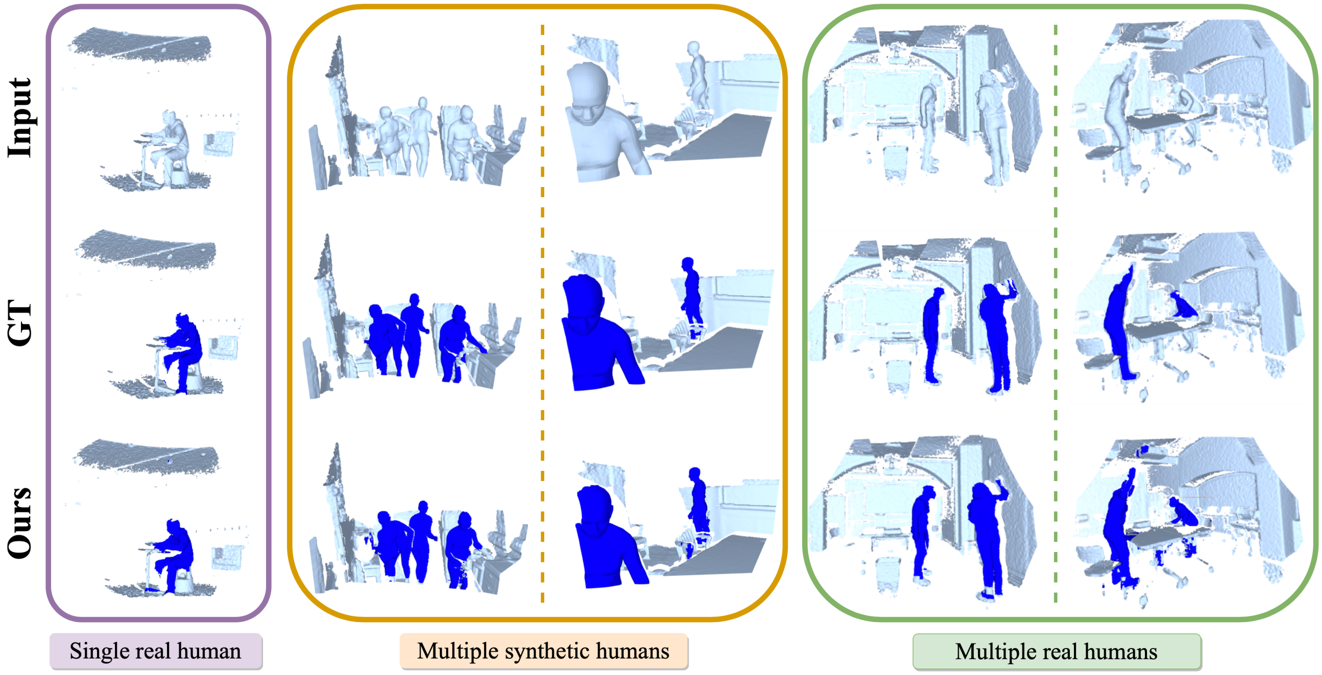

Figure 1. Given challenging indoor depth scans (top row) of humans interacting with their surroundings and with each other, our method is capable of segmenting humans (bottom row) realistically and accurately. We train and evaluate on scenes with **a single real human**, **multiple synthetic humans**, and **multiple real humans**.

Abstract

Segmenting humans in 3D indoor scenes is becoming increasingly important with the rise of human-centered artificial intelligence that tries to perceive human-scene interactions or social interactions between multiple humans. However, current methods for 3D semantic segmentation mostly focus on objects. This is largely due to the nature of existing 3D indoor datasets since they contain hardly any annotated humans. Moreover, other datasets that focus on human-object interactions exhibit limited diversity in terms of human poses and occlusion patterns which limits generalization to unseen scenes. In this work, we propose a pipeline to augment 3D indoor datasets with synthetically generated humans, as well as real human scans, which results in datasets that cover a large variety of human poses and scene interactions. We also devise a method for segmenting humans in depth scans rendered from the populated 3D scenes and provide an in-depth study of the generalization performance of our models across different scene settings. Furthermore, we show that it is key to jointly train on real and synthetic data and report a significant improvement over models trained on a single modality. Our code is available for research purposes at https://github.com/aycatakmaz/segment-humans-3d.

1. Introduction

Semantic segmentation of 3D scenes has seen tremendous progress over the last years [9, 13, 16, 19, 24–26, 28, 33, 37, 38, 40]. While current methods for 3D indoor scenes achieve strong performance across many settings, they are largely limited to segmenting scene objects, such as furniture and windows. On the other hand, the ability to segment humans in 3D scenes is becoming increasingly important as the need for human-centered datasets emerges and 3D scanners become more available. Providing a method that accurately segments humans in 3D scenes can be beneficial for many applications, such as augmented reality interactions, generation of 3D human-scene interaction datasets, and motion capture in complex scenes. However, this fundamental topic has been rather under-explored through the lens of existing 3D semantic segmentation approaches.

We argue that one of the limiting factors in learning to segment humans in 3D scenes is the limited availability of annotated 3D datasets that include humans interacting with the scenes and with each other. While most autonomous driving datasets for 3D semantic segmentation [2, 4, 29, 36] include humans, these outdoor scenarios often have a limited variety in human poses and actions, such as standing or walking. We are particularly interested in general human-scene interaction scenarios capturing a broader variety of human poses and actions, which often take place in *indoor* settings. However, existing large-scale indoor datasets with segmentation annotations, such as Matterport3D [5] and ScanNet [10], either completely lack humans or only have a few human instances. Although a few steps towards more diverse interaction datasets have recently been taken with BEHAVE [3] and EgoBody [41], none of them has been used for learning to segment humans thus far.

In this work, we explore the task of segmenting humans in partial point clouds obtained from depth scans of scenes in which humans are interacting with their surroundings and with each other. We focus on indoor settings where we can potentially have one or more humans engaging in activities that involve scene objects. As we argue that one of the main limitations for this task is the limited availability of indoor scenes with annotated humans, we first focus on populating 3D indoor scenes with synthetic and real humans. We augment the ScanNet [10] dataset with synthetic humans using PLACE [42] and with real human scans from BEHAVE [3]. We then render frames from the populated 3D scenes to simulate a depth camera. We train our models on partial point clouds obtained from these depth scans, as well as scenes from BEHAVE [3]. We also jointly train our models on real and synthetic data and show that they outperform the models trained on either real or synthetic data alone. We perform experiments to cross-examine the generalization capability of our models trained on different sets of training data by testing them on scenes from other datasets with varying scene settings. As shown in Figure 1, our models trained with scenes including a single real human, multiple synthetic humans and multiple real humans are capable of segmenting humans realistically and accurately. Our contributions are as follows:

1. We focus on the very important yet largely ignored task of segmenting humans in depth scans of complex, cluttered scenes and propose a method for this task that is trained on synthetic, real, and joint data.

2. We propose a pipeline to augment 3D indoor datasets with synthetically generated humans, as well as real human scans, interacting with their surroundings and render depth frames from the populated 3D scenes to simulate a depth camera.

3. We conduct an extensive set of experiments that provide insights about the generalization performance of our model across different settings and show that the model jointly trained on synthetic and real data outperforms the models trained on a single modality.

2. Related Work

2.1. Segmenting 3D Scenes

The goal of 3D semantic segmentation is to assign a semantic label to each point in a given 3D scene. Before the emergence of deep learning based methods, the classical methods formulate this problem as a graphical model and combine it with a classifier stage [12, 20, 21]. With the increased availability of annotated 3D datasets and the tremendous advancements in the performance of deep neural networks, numerous methods have been proposed for the task of 3D semantic segmentation. These methods can be divided into two groups: voxel-based methods [9, 13, 16, 25, 37] and point-based methods [11, 22, 23, 31, 32, 34].

Voxel-based methods consist of two main operations: transforming unstructured 3D point clouds into regular volumetric grids (voxels) and applying 3D convolutions to perform semantic segmentation. The initial work relies on using fully convolutional neural networks (FCNNs) [25]. In [16], Huang *et al.* proposed to use 3D-FCNN for the classifier stage in order to perform voxel-level semantic labeling. This work was extended in [37] by feeding the coarse voxel-level semantic labels to a trilinear interpolation layer and applying a 3D fully connected CRF to obtain fine-grained semantic segmentation.

More recent voxel-based methods rely on sparse convolutional networks [9, 13]. In [13], Graham *et al.* proposed submanifold sparse convolutional networks (SSCNs) for processing sparse 3D point clouds and performing semantic segmentation. SSCNs are shown to outperform the state-of-the-art approaches both in segmenting objects in large 3D scenes and in segmenting parts of objects in small 3D scenes. In [9], Choy *et al.* proposed the Minkowski Engine which is an auto-differentiation library for sparse tensors. As it supports a wide range of neural network layers, it is used as a building block in many applications, such as segmentation and classification. Following many recent indoor segmentation methods, we rely on the Minkowski Engine as our model backbone.

Point-based methods operate directly on 3D point clouds without imposing structure which improves computation time and reduces artifacts caused by voxelization. These methods became popular with PointNet [31] which uses a max-pooling layer to extract global features and fuses them with point features obtained through a sequence of MLPs before assigning semantic labels to objects. However, this network does not capture local features which are crucial for semantic segmentation. In [32], Qi *et al*. applied PointNet [31] hierarchically on nested groupings of input points to learn local features. There are also other deep learning-based frameworks [11, 22, 23, 34] that improve the performance of existing point-based methods.

2.2. Populating 3D scenes with realistic humans

Although the explained methods achieve state-of-the-art results in segmenting objects in 3D scenes, none of them focuses on segmenting humans in indoor scenes. This may be due to the limited availability of 3D indoor datasets with humans. For this reason, methods for populating 3D scenes with realistic humans interacting with the environment have recently emerged. In PLACE [42], Zhang *et al*. proposed to train a conditional VAE to learn plausible proximal relationships between a human body and a 3D scene, generate a full 3D body mesh that comply with the predicted human-scene proximity, and refine the body mesh through optimization. This method does not use semantics, but it encodes proximal relations and the scene shape with Basis Point Sets (BPS) to synthesize naturalistic 3D human bodies, represented by the SMPL-X model [30]. In POSA [15], Hassan *et al*. proposed to extend the SMPL-X model [30] to encode the contact probability with the surface and the corresponding semantic label for every mesh vertex. This method uses the scene mesh and the extended SMPL-X body mesh [30] to place realistic humans in 3D scenes using the semantics.

2.3. Segmenting humans in depth scans

Considering that it is not always practical to acquire full scans of scenes in real world environments, various methods have been proposed for detecting humans using depth images [8, 39] and segmenting humans or human parts in depth scans [17, 18, 39]. In [8], Choi *et al*. proposed an algorithm to detect humans in indoor settings using depth images obtained from an RGB-D camera. This algorithm uses a graph-based segmentation method to segment depth images, applies parameterized heuristics to obtain a set of candidates, and computes a depth-based descriptor for each candidate to detect humans. In [39], Xia *et al*. proposed a model-based human detection method using depth images obtained from a Kinect camera and a segmentation method to separate the detected humans from the background. These methods are shown to detect and extract the contours of humans accurately on real 3D Kinect sequences.

In [18], Jalal *et al*. proposed an approach to segment and track human actions using depth images obtained from a Kinect camera. This approach extracts the human silhouettes from the background on the depth images and applies two other mechanisms to track actions. In [17], Hynes *et al*. proposed to segment human parts from depth images using a graph-based approach which uses the image positions of each part as a prior. Although segmenting humans from depth scans is not a new task, it is largely ignored compared to segmenting 3D scenes. Moreover, the current methods mostly consider uncluttered environments containing a single human which is not occluded. Hence, we focus on segmenting humans in depth scans of complex environments in which multiple humans are interacting with each other and with their surroundings and are potentially occluded.

2.4. Available 3D datasets

Existing 3D datasets significantly differ in terms of their modalities, as well as their targeted tasks. There are datasets of 3D objects, such as PartNet [27] and ShapeNet [6]. There are also several datasets of 3D scenes with semantic segmentation annotations, such as ScanNet [10] Matterport3D [5], and Replica [35]. However, these datasets either completely lack humans or only have a few human instances.

PROX-E [14, 43] is a 3D indoor scene dataset with semantic scene annotations in which humans, represented by the SMPL-X [30] model, interact with their surroundings. However, this is a small-scale dataset which does not provide sufficient data to train a segmentation model that can handle a large number of interaction scenarios. The recently released BEHAVE dataset [3] provides multi-view RGB-D frames containing humans interacting with objects, but it has not been used for 3D semantic segmentation tasks yet.

There are also several autonomous driving datasets containing humans, such as [2, 4, 29, 36]. However, they are not suitable for our task for a variety of reasons. Firstly, outdoor scenes differ from indoor scenes in terms of the objects present in the scene and their arrangements. Secondly, humans are underrepresented compared to other classes, such as roads and cars. Finally, human poses and their interactions with objects are less varied in outdoor scenes.

3. Method

In this work, we firstly target the limited availability of datasets containing humans interacting with each other and with their surroundings. This section provides an overview of our approach for existing datasets, such as BEHAVE [3], as well as the methods we propose to obtain depth scans of 3D scenes populated with synthetic and real humans.

3.1. Datasets

BEHAVE [3]. The BEHAVE dataset [3] is a large-scale dataset of human-object interactions captured in natural environments. It provides 3D human, object, and contact annotations. The interactions between the subjects and objects are captured using 4 Kinect RGB-D cameras. Each frame contains human and object masks together with segmented point clouds. We obtain partial point clouds using the depth

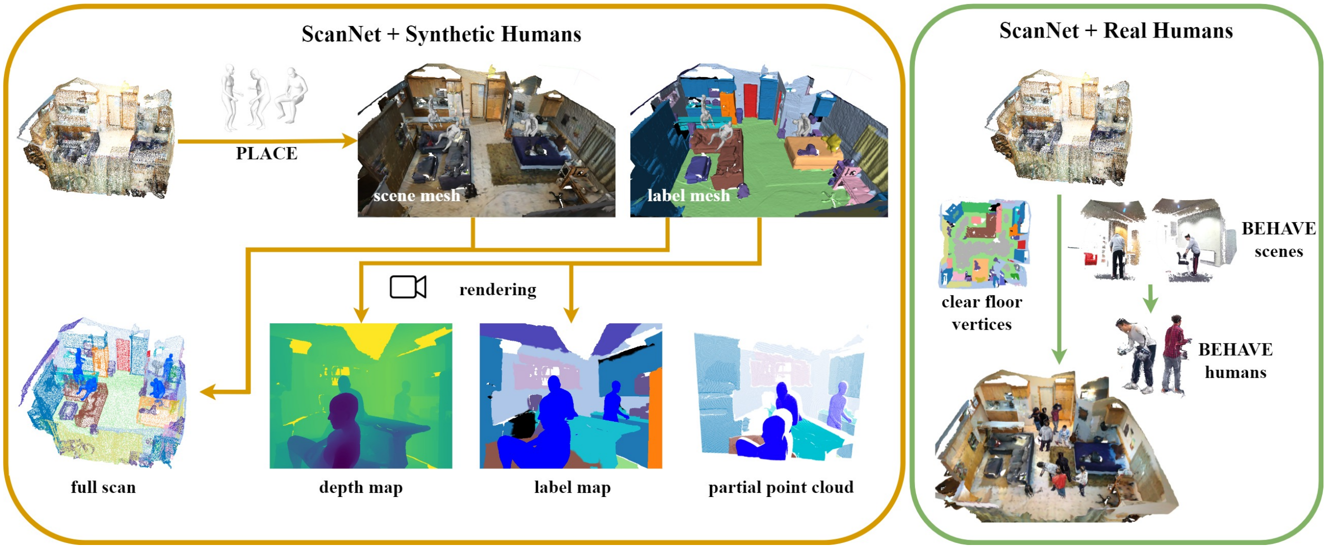

Figure 2. Given a scene mesh from ScanNet [10], we populate it with **synthetic humans** using PLACE [42]. From the resulting scene with humans, we obtain two sets of training data: firstly, we combine it with the label mesh to obtain **full point clouds** and secondly, we render it to acquire depth and label maps which are then backprojected to obtain **partial point clouds**. For placing **real humans**, we find the clear floor vertices, i.e. floor vertices that do not have any objects on or near them, of the given scene and place BEHAVE [3] humans on vertices sampled from them. The rest of the data generation process is the same as explained for scenes with synthetic humans.

images provided in the dataset and pre-process each point cloud to create a point cloud segmentation dataset for training. Since the point clouds are very dense ($\sim$ 800k points per scene), we sample 20% of points for each scene. We use the originally provided train-validation-test split and obtain $\sim$41k point clouds as training samples, $\sim$5k validation samples, and $\sim$18k test samples in total from all frames captured from each Kinect camera.

Although BEHAVE [3] is the largest dataset to this date to provide 3D annotations for human-object interactions, it suffers from multiple limitations. Firstly, the dataset is limited to only 8 subjects and 20 objects. Moreover, the human and object masks registered from multi-views are rather noisy and sometimes inaccurately reconstructed, especially in the presence of interaction objects with which humans are in close contact (e.g. backpacks). Furthermore, each frame contains a single human, always at a minimum certain distance, captured from a Kinect camera located at one of the edges of the room. For more realistic capturing scenarios, e.g. with a hand-held mobile camera with a depth sensor, this dataset is potentially not representative, and a model trained with the BEHAVE dataset might not generalize well. In the following subsections, we address this problem and explain how we populate scenes and capture depth maps of more realistic scenarios, which can have a more practical value in potentially complex indoor scenes.

ScanNet [10] populated with synthetic humans. The ScanNet dataset [10] is a dataset of RGB-D scans of real-world environments. It provides 2.5 million RGB-D images acquired in 1513 scenes and 707 rooms and annotated with semantic segmentations, surface reconstructions, and 3D camera poses. However, it only contains a few human instances. In the light of our previous discussion about our aim to obtain more realistic human-scene interaction scenarios, we augment the ScanNet dataset [10] by placing synthetic humans in realistic poses using PLACE [42]. To achieve this, we first calculate the signed distance field (SDF) and find the scene boundaries for all training scenes. The SDF value is 0 on the surfaces or boundaries of a set, so PLACE [42] uses the scene SDFs and the scene boundaries to find suitable surfaces to place synthetic humans.

Since we want to capture a variety of human poses and actions, we modify PLACE [42] to perform instance segmentation guided human location sampling. For this purpose, we extract the coordinates and labels of all objects in all training scenes and use this information for generating bounding boxes of $2m^3$ to place humans. The size for the bounding boxes is selected as the same size used in the training of PLACE [42]. We generate a maximum number of 10 synthetic humans per scene where the maximum number depends on the number of objects present in the scene. During generation, we give priority to some objects, such as tables and chairs, to capture different human poses and then randomly choose among other objects if these do not exist in the scene or the maximum number of people has not been reached yet. We also use 200 and 100 iterations for simple and advanced optimization of PLACE [42], respectively. Moreover, we increase the weight of the collision loss term (from 8.0 to 10.0) in advanced optimization to reduce interpenetrations, but there are still some failure cases around thin structures, e.g. tables and chairs. We save optimized body meshes along with optimized body parameters and vertices for each scene.

We then select a random number of humans $n_{human} \in \{0, 1, \ldots, N_{max}\}$ for each scene and save their indices. The pre-processing of the scenes includes combining the low resolution scene meshes and the low resolution scene labels to produce 3D point clouds where each point is annotated with GT semantic category. During data-loading, we load n_{human} selected humans and append their point clouds to the scene point cloud if n_{human} is not zero.

ScanNet [10] populated with real humans. In this approach, we augment the ScanNet dataset [10] with real human scans provided in the BEHAVE dataset [3]. For this purpose, we extract clear floor vertices from all training scenes by using the low resolution scene labels and save them. We set a safety threshold distance of 0.5 meters and filter all floor vertices such that if there is a vertex within the safety threshold distance that is not floor in the xy-plane, that floor vertex is not clear. This eliminates the floor vertices at the edges of the scene and the floor vertices that have some objects on or near them. This also ensures that there is sufficient space for the human to be placed.

We then select a random number of humans $n_{human} \in \{0, 1, \ldots, 10\}$ from BEHAVE [3] for each scene. During data-loading, we load the scene and n_{human} selected humans for that scene. For each scene, we randomly sample n_{human} clear floor vertices and apply transformations to place the humans to the selected locations. Since each loaded human point cloud is upside down, we first transform it to be upright and then translate it to the desired vertex location. We also find the smallest z coordinate of each human point cloud and translate it such that the humans are always placed on the floor. We then append the human point clouds to the scene point cloud if n_{human} is not zero.

Rendering RGB-D scans from full scans. As humans are dynamic in nature and it is not practical to acquire full scans of scenes with humans during real testing environment, we focus on depth scans for our pipeline. With this in mind, we aim to simulate real testing environments in order for our work to have a higher practical value. In that direction, we render RGB-D images from full scene meshes with synthetic humans acquired by PLACE [42] and with real humans from BEHAVE [3]. We use label meshes of ScanNet [10] and place synthetic or real humans in them as explained before. An artificial camera is then placed at the center of each scene with a height uniformly sampled between [1.40, 1.60] meters. The camera is always aligned with the ground (i.e. xy-plane), but it looks at a random direction sampled between [0, 360) degrees. For each scene, 30 RGB-D images are rendered with camera's height and direction resampled at each iteration.

Throughout the paper, we refer to the RGB-D dataset obtained from ScanNet [10] scenes populated via PLACE [42] as *ScanNet + PLACE* and to the RGB-D dataset obtained from ScanNet [10] scenes populated with real human scans from BEHAVE [3] as *ScanNet + BEHAVE*. Both ScanNet + PLACE and ScanNet + BEHAVE datasets include ∼45k frames each and are provided with ground truth semantic label maps featuring the same set of semantic categories from ScanNet [10]. Please note that we only use the depth maps and the respective semantic label maps in our experiments.

Kinect RGB-D interaction sequences. To assess how our model performs on real world data captured in more realistic and complex settings, we use several Kinect RGB-D sequences from the EgoBody [41] dataset. There are 4 different recordings that capture interactions between two subjects in an indoor environment. Each recording is captured from a main Kinect camera as well as 4 other Kinect cameras, i.e. for each interaction recording, there are 5 RGB-D sequences in total. In order to extract pseudo-ground truth segmentation annotations, each frame is passed through a pre-trained Mask-RCNN model [1] and the predictions are post-processed to obtain binary masks for human bodies in each RGB image followed by a morphological opening operation on the masks. Please note that the pseudo-ground truth can be noisy, hence the point cloud annotations are noisy in some frames. By back-projecting the depth images, we obtain a point cloud for each scene, annotated with pseudo-ground truth binary labels: human or background.

4. Experiments

In this section, we first provide details about our model architecture as well as the training procedure. We then explain different experiment settings we considered, and report the performance of each model. We additionally perform experiments to cross-examine the generalization capability of our models trained on different sets of training data by testing them on scenes from other datasets with different scene settings. Through our analysis, we demonstrate that our models trained with a single real human, multiple synthetic humans and multiple real humans are capable of segmenting humans realistically and accurately, given that input point clouds meet appropriate scene priors.

4.1. Network and training details

For our backbone architecture, we employ a U-Net variant of the Minkowski Engine [9], Res16UNet34A. We train with a stochastic gradient descent optimizer with momentum 0.9 and dampening factor 0.1. We employ weight decay with a decaying coefficient of 10^{-4}. Scheduling of the learning rate is done via poly learning rate policy [7] (for details, see supplementary material). We limit the maximum number of points to 1.2M, and sub-sample points before voxelizing the scene when necessary. We train with elastic distortion augmentation as well as odd/even coordinate augmentation to make the model more robust.

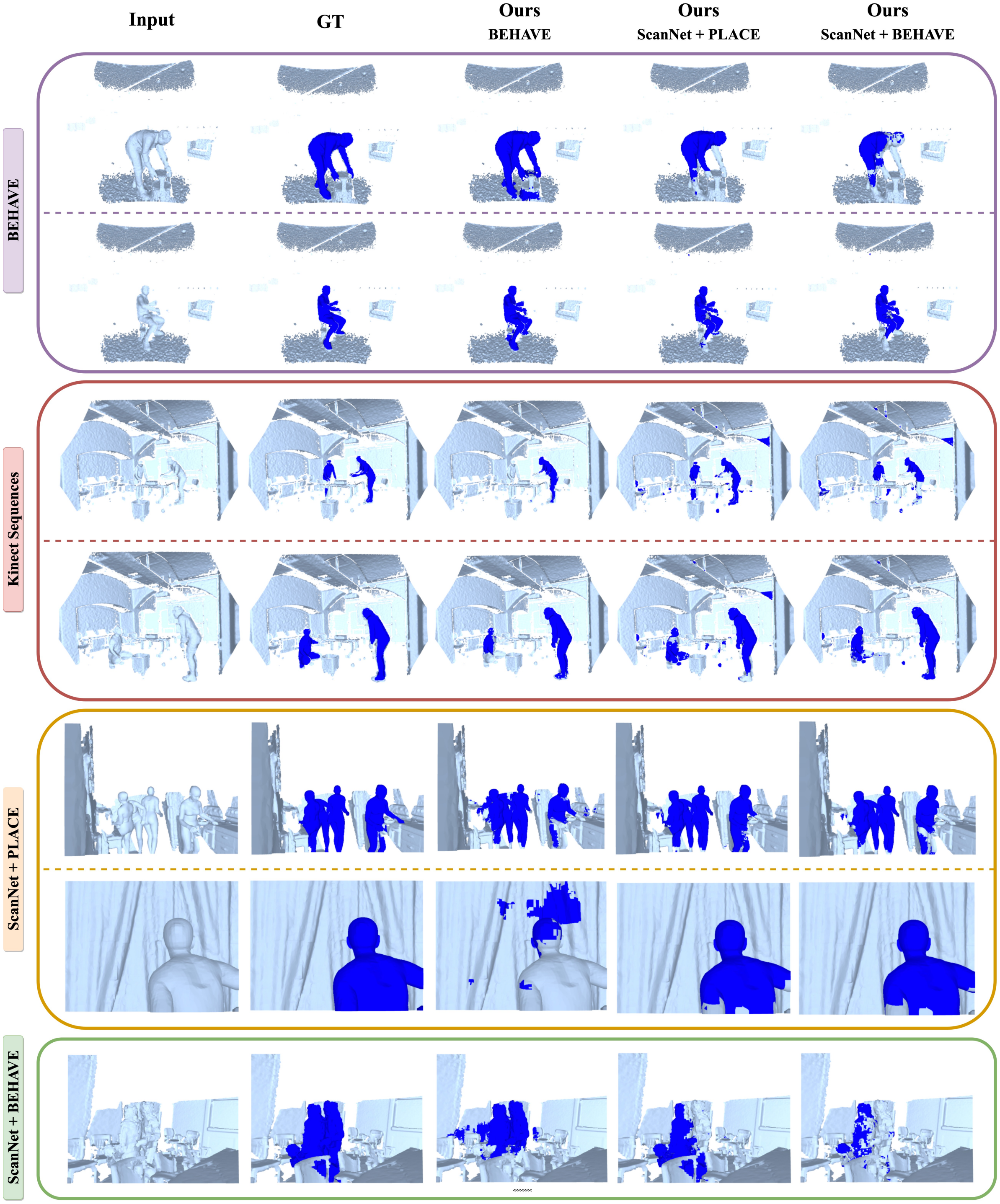

Figure 3. Example predictions and comparisons. From left to right, we show: (1) input point cloud, (2) ground truth segmentation, result from (3) model trained with the BEHAVE dataset, (4) model trained with depth scans from ScanNet scenes populated with PLACE, (5) model trained with depth scans from scenes populated with BEHAVE, using two labels **person** and **background**. Each box denotes results on a different dataset: **BEHAVE**, **Kinect RGB-D Sequences**, **ScanNet+PLACE** and **ScanNet+BEHAVE**. We demonstrate successful results on challenging scenes with one or more humans, often occluded and closely interacting with their surroundings. Bottom two rows demonstrate failure cases for different methods: first one is a close-up view of a human. While models trained with ScanNet+PLACE and ScanNet+BEHAVE perform well on this point cloud, the model trained with BEHAVE fails, whose training only involves humans with fully visible bodies. Last row demonstrates a scene with very noisy human bodies, which is challenging for our models to handle.

Dataset	Scene	Voxel Size	Model
BEHAVE	depth scan	2 cm	Res16UNet34A
ScanNet + PLACE	full scan	2 cm	Res16UNet34A
ScanNet + PLACE	depth scan	2 cm	Res16UNet34A
ScanNet + BEHAVE	full scan	2 cm	Res16UNet34A
ScanNet + BEHAVE	depth scan	2 cm	Res16UNet34A

Table 1. MinkowskiNet [9] architectures and training settings. We provide architectural details about the networks as well as input configurations for training the pipeline for different datasets. We use standard MinkowskiNet architectures with Res16UNet34A backbones that are available in the official Minkowski repository.

Unlike recent 3D full scan/RGB-D approaches for semantic segmentation, our input features do not include RGB color features. Instead, we use a constant feature value of 1 for each voxel. The main reason behind this is that we use several methods to populate our scenes with realistic humans whereas these methods either do not provide realistic textures for the humans or they do not consider constraints, such as environment lighting, to assign realistic RGB values for each point. Therefore, we opted out from using the RGB data and decided to solely rely on the scene geometry.

For our experiments using the BEHAVE dataset, we utilize weighted cross-entropy in order to account for the class imbalance in the dataset. For all other experiments, we use cross-entropy as our training objective. Our pipeline takes voxelized scans of 3D scenes and assigns a semantic class to each voxel. Predictions for points are obtained via label propagation based on the closest voxel center.

4.2. Evaluation

Evaluation metrics. We evaluate the performance of our model by employing commonly used evaluation metrics for 3D semantic segmentation, following previous works. In our quantitative evaluations, we report the mean intersection-over-union *(mIoU)*, mean average precision *(mAP)* and mean class accuracy *(mAcc)*.

4.3. Experiments

We train our semantic segmentation models with real scenes from BEHAVE [3], synthetic scenes from ScanNet [10] populated with humans, as well as the combination of these two datasets in a joint training setting. In all of our experiments we use a voxel size of 2 cm.

Training with the BEHAVE dataset. We train our model with partial point clouds obtained using the depth maps from the BEHAVE dataset [3]. As the BEHAVE dataset provides dense annotations in the image level for humans, interaction objects and background, there are different potential strategies to formulate the problem. In our experiments, we map all classes outside of the *human* class to *background* category and train a semantic segmentation model with two classes: *human* or *background*.

Training with ScanNet scenes populated with humans. For our experiments with the ScanNet dataset, we follow two approaches to populate the scenes with realistic humans interacting with their surroundings, as outlined in Section 3.1. In the first approach, we use PLACE [42] to populate the scenes using synthetic humans. In the other approach, we use human point clouds provided in the BEHAVE dataset. For both of these approaches, we define the set of labels depending on the training input type. For our model trained with full scans of scenes populated with humans, we train with 20+1 labels, where we add the *person* category to the original 20 categories commonly used in the literature for training semantic segmentation models on the ScanNet dataset [10]. For our model trained with depth scans obtained via rendering (see 3.1), we train with binary labels, namely *human* and *background*. One of the main reasons for this design choice is our empirical observation that using a small number of renders (∼30) from each scene does not capture the variety of semantic categories in the dataset, which in turn results in poor performance. Instead, for the case with depth scans, we focus our efforts on segmenting humans from the background. As explained earlier, we are particularly interested in the scenario with depth scans due to its practical significance.

Training data	Input	Voxel Size	mIoU ↑	mAP ↑	mAcc ↑
BEHAVE	depth	2 cm	**94.1**	**99.2**	**97.6**
ScanNet + PLACE	depth	2 cm	66.5	91.0	71.8
ScanNet + BEHAVE	depth	2 cm	66.5	92.5	71.8

Table 2. Results on the test split of the BEHAVE dataset [3]. We obtain the best performance using the model trained with BEHAVE. In the ScanNet renders, humans are often closer to the camera and their bodies are visible only partially unlike BEHAVE scenes. We observe that models trained with ScanNet renders performs worse than BEHAVE models on BEHAVE scenes, due to different scene priors.

ScanNet + PLACE					
Training data	Input	Voxel Size	mIoU ↑	mAP ↑	mAcc ↑
BEHAVE	depth	2 cm	57.8	86.5	66.0
ScanNet + PLACE	depth	2 cm	**93.1**	**97.9**	**95.6**
ScanNet + BEHAVE	depth	2 cm	87.6	97.4	90.4
ScanNet + BEHAVE					
Training data	Input	Voxel Size	mIoU ↑	mAP ↑	mAcc ↑
BEHAVE	depth	2 cm	55.2	88.3	61.1
ScanNet + PLACE	depth	2 cm	**57.4**	91.8	**61.3**
ScanNet + BEHAVE	depth	2 cm	53.0	**92.4**	56.9

Table 3. Results on the depth scans from ScanNet [10] scenes populated using PLACE [42] as well as populated with humans from the BEHAVE [3] dataset. In most metrics and test scenarios, the model trained with ScanNet+PLACE renders performs the best.

Evaluation on Kinect RGB-D interaction sequences. In addition to our experiments with the BEHAVE and ScanNet datasets, we also assess how our model performs on real world data captured in more realistic and complex settings. For this purpose, we utilize the Kinect RGB-D interaction sequences previously described in Section 3.1 in order to evaluate how well our model generalizes to unseen, realistic scenarios in which multiple humans are closely interacting with each other as well as their surroundings. While we report quantitative results from our model using the pseudo-ground truth labels in Table 4, we also assess the model performance qualitatively in Figure 3. We evaluate on the models trained on real scenes from BEHAVE, synthetic scenes populated with humans, and with the combination of these two modalities. In Table 4, we refer to the model that was trained with real scenes *and* synthetic scenes as *joint*.

Training data	Input	Voxel Size	mIoU ↑	mAP ↑	mAcc ↑
BEHAVE	depth	2 cm	78.1	82.7	83.8
ScanNet + PLACE	depth	2 cm	70.0	84.9	76.7
ScanNet + BEHAVE	depth	2 cm	63.6	81.4	68.7
Joint	depth	2 cm	**82.9**	**89.6**	**88.9**

Table 4. Results on the EgoBody [41] Kinect RGB-D sequences. Model jointly trained on real and synthetic scenes performs the best on the Kinect sequences. Among the other models trained with only one modality, training on the BEHAVE dataset performs better compared to training with synthetic scenes. This is potentially due to the fact that the Kinect sequences are of similar nature as the BEHAVE scenes in terms of the positioning of humans with respect to the camera.

Analysis of results. As shown in Table 2, we observe that the model trained on BEHAVE performs the best when tested on the BEHAVE test set. General scene settings in the BEHAVE dataset are different than the renders from the populated ScanNet scenes in terms of the number of humans present, as well as the positions of humans with respect to the camera. In the ScanNet renders, humans are often closer to the camera and their bodies are visible only partially unlike BEHAVE scenes. We observe that models trained with ScanNet renders performs worse than BEHAVE models on BEHAVE scenes, due to different scene priors.

In Table 3, it can be seen that the model trained with renders from ScanNet+PLACE results in superior performance on most metrics, both on the ScanNet+PLACE test set as well as the ScanNet+BEHAVE test set. An important thing to note that the ScanNet+PLACE dataset is a dataset with synthetically created clean annotations whereas the human point clouds in the BEHAVE dataset are registered from 4 Kinect views and are quite noisy. Therefore, we suspect that one of the main reasons why the model trained using the renders from scenes populated with BEHAVE humans underperform on almost all experiments, including the evaluations on ScanNet+BEHAVE renders, is the noise during the training process. This finding highlights that even when we are placing more realistic humans compared to SMPL models, unclean labels hinder the training process.

Finally, in Table 4, it can be seen that the model trained with the BEHAVE dataset has superior performance compared to the models trained with renders from the populated 3D scenes. We suspect that one reason for this is the similarity between the scene settings in the BEHAVE dataset and the Kinect sequences, both of which were captured using Kinect cameras placed at one of the edges of the room and from a certain distance from the subjects. This finding highlights the importance of selecting a suitable model based on the human-camera positioning prior. Furthermore, we have trained another model where we used samples from BEHAVE dataset as well as samples from ScanNet+PLACE simultaneously during training, which is the *joint* training setting. As shown in Table 4, our model trained jointly on real and synthetic scenes has significantly better performance compared to the models which were trained on only real or only synthetically populated scenes. Our findings show that it is *key* to jointly train on real and synthetic data.

5. Discussion

5.1. Limitations and future work

An important future direction is to explore the placement of realistic humans with clothes. Another aspect is that we currently only utilize the depth data and we do not use RGB colors as features. This enables us to place textureless SMPL body models in our scenes, which makes it easier to increase the scale of the variety in body shapes and poses. In later work, we will focus on the generation of more realistic synthetic data, ideally placing clothed humans instead of directly using the SMPL model.

5.2. Conclusion

In this work we focus on segmenting humans interacting with indoor scenes. We identify that one of the limiting factors in this field is the limited availability of large-scale, annotated 3D datasets with humans. To address this, we utilize methods to populate 3D scenes from ScanNet [10] with synthetic humans as well as scans of real humans, and we collect a new semantic segmentation dataset of 3D scenes in which humans are closely interacting with their surroundings. By capturing depth maps from the populated ScanNet scenes, we perform extensive quantitative and qualitative evaluations that demonstrate the benefits of training a semantic segmentation model on 3D scenes that were populated with realistic humans. Furthermore, we analyze the generalization capability of our models trained with scenes with different human-camera priors when evaluated on different sets of data. Through our evaluations on real scenes, we demonstrate that the key is to jointly train on real and synthetic data instead of only using synthetic data.

6. Contributions of team members

Cafer and İrem were primarily responsible for placing synthetic and real humans in ScanNet scenes, as well as capturing depth scans from the scenes populated with humans to synthetically create a human segmentation dataset. Ayça was primarily responsible for implementing the training pipeline with the MinkowskiEngine and evaluating the human segmentation performance. Workload for all other tasks was distributed equally among the team members.

References

[1] Waleed Abdulla. Mask r-cnn for object detection and instance segmentation on keras and tensorflow. https://github.com/matterport/Mask_RCNN, 2017. 5

[2] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*, 2019. 2, 3

[3] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2022. 2, 3, 4, 5, 7

[4] 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 *CVPR*, 2020. 2, 3

[5] 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. *International Conference on 3D Vision (3DV)*, 2017. 2, 3

[6] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 3

[7] 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. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 5

[8] Benjamin Choi, Çetin Meriçli, Joydeep Biswas, and Manuela Veloso. Fast human detection for indoor mobile robots using depth images. In *2013 IEEE International Conference on Robotics and Automation*, pages 1108–1113, 2013. 3

[9] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 1, 2, 5, 7

[10] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 2, 3, 4, 5, 7, 8

[11] Francis Engelmann, Theodora Kontogianni, Alexander Hermans, and Bastian Leibe. Exploring spatial context for 3d semantic segmentation of point clouds. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017. 2, 3

[12] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004. 2

[13] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. *CoRR*, abs/1711.10275, 2017. 1, 2

[14] Mohamed Hassan, Vasileios Choutas, Dimitrios Tzionas, and Michael J. Black. Resolving 3D human pose ambiguities with 3D scene constraints. In *International Conference on Computer Vision*, Oct. 2019. 3

[15] Mohamed Hassan, Partha Ghosh, Joachim Tesch, Dimitrios Tzionas, and Michael J. Black. Populating 3D scenes by learning human-scene interaction. In *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 3

[16] Jing Huang and Suya You. Point cloud labeling using 3d convolutional neural network. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 2670–2675, 2016. 1, 2

[17] Andrew Hynes and Stephen Czarnuch. Human part segmentation in depth images with annotated part positions. *Sensors*, 18(6), 2018. 3

[18] 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. *Journal of Electrical Engineering and Technology*, 14:455–461, 01 2019. 3

[19] Abhijit Kundu, Xiaoqi Yin, Alireza Fathi, David Ross, Brian Brewington, Thomas Funkhouser, and Caroline Pantofaru. Virtual multi-view fusion for 3d semantic segmentation. In *European Conference on Computer Vision*, pages 518–535. Springer, 2020. 1

[20] Lubor Ladicky, Chris Russell, Pushmeet Kohli, and Philip H. S. Torr. Associative hierarchical random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1056–1077, 2014. 2

[21] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. 2

[22] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. *CoRR*, abs/1711.09869, 2017. 2, 3

[23] Yangyan Li, Rui Bu, Mingchao Sun, and Baoquan Chen. Pointcnn. *CoRR*, abs/1801.07791, 2018. 2, 3

[24] Kangcheng Liu, Zhi Gao, Feng Lin, and Ben M Chen. Fg-net: Fast large-scale lidar point clouds understanding network leveraging correlated feature mining and geometric-aware modelling. *arXiv preprint arXiv:2012.09439*, 2020. 1

[25] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1411.4038, 2014. 1, 2

[26] Hsien Yu Meng, Lin Gao, Yu Kun Lai, and Dinesh Manocha. Vv-net: Voxel VAE net with group convolutions for point cloud segmentation. *CoRR*, abs/1811.04337, 2018. 1

[27] Kaichun Mo, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3

[28] 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. 1

[29] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *International Conference on Computer Vision (ICCV)*, 2017. 2, 3

[30] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 10975–10985, 2019. 3

[31] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *CoRR*, abs/1612.00593, 2016. 2, 3

[32] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *CoRR*, abs/1706.02413, 2017. 2, 3

[33] Haoxi Ran, Wei Zhuo, Jun Liu, and Li Lu. Learning inner-group relations on point clouds, 2021. 1

[34] Martin Simonovsky and Nikos Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. *CoRR*, abs/1704.02901, 2017. 2, 3

[35] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 3

[36] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott M. Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2443–2451, 2020. 2, 3

[37] Lyne P. Tchapmi, Christopher B. Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. Segcloud: Semantic segmentation of 3d point clouds. *CoRR*, abs/1710.07563, 2017. 1, 2

[38] Li Jiang Jiaya Jia Wenbo Hu, Hengshuang Zhao and Tien-Tsin Wong. Bidirectional projection network for cross dimensional scene understanding. In *CVPR*, 2021. 1

[39] Lu Xia, Chia-Chih Chen, and J. K. Aggarwal. Human detection using depth information by kinect. In *CVPR 2011 WORKSHOPS*, pages 15–22, 2011. 3

[40] Qiangeng Xu, Xudong Sun, Cho-Ying Wu, Panqu Wang, and Ulrich Neumann. Grid-gcn for fast and scalable point cloud learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5661–5670, 2020. 1

[41] Siwei Zhang, Ma Qianli, Yan Zhang, Qian Zhiyin, Pollefeys Marc, Federica Bogo, and Siyu Tang. Egobody: Human body shape, motion and social interactions from head-mounted devices. *arXiv preprint arXiv:2112.07642*, 2021. 2, 5, 8

[42] Siwei Zhang, Yan Zhang, Qianli Ma, Michael J Black, and Siyu Tang. Place: Proximity learning of articulation and contact in 3d environments. In *2020 International Conference on 3D Vision (3DV)*, pages 642–651. IEEE, 2020. 2, 3, 4, 5, 7

[43] Yan Zhang, Mohamed Hassan, Heiko Neumann, Michael J. Black, and Siyu Tang. Generating 3d people in scenes without people. In *Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3