A. Implementation Details

In this section, we first provide detailed ground removal implementation, then we describe how we insert synthetic human. Next, we give details of our Bi-directional tracking filter. After that, we describe how we select key joints. At last, we provide our training settings.

Ground removal. Given a LIDAR point cloud, we employ RANSAC [2] with additional refinement and constraints to segment the ground point cloud. First, we partition the detection range of the point cloud into 2-dimensional patches, *e.g.* $5m \times 5m$. Then for each patch, we voxelize the point cloud within this patch, then run RANSAC with a threshold of 0.06 in the lowest voxels (force RANSAC to choose points randomly only in these voxels) to obtain each patch's ground point cloud. The voxel size we use here is $[0.1, 0.1, 0.05]$. Specifically, we add following constrains on RANSAC:

- The fitted plane is required to exhibit an angle with the xy-plane that is less than 25 degrees.
- The fitted plane should contain at least 50 points.
- The quantity of points below the fitted plane should be less than 20% of the total points on the plane.
- The mean distance of points below the fitted plane from the plane itself is less than 0.15 meters.

If all these conditions are satisfied, we rerun RANSAC with these conditions 6 times more and combine the result as the final ground point cloud in this patch. Finally, after we obtain all patches' ground point cloud, we combine them as the scene's ground point cloud.

Synthetic human insertion. We first choose a random sequence in the dataset, then choose a random frame within this sequence. Leveraging the segmented ground point cloud, we choose a random distance in the obtained ground point cloud for this frame. Then randomly choose one point from all the points that satisfy this distance as the initial insertion location. In detail, we first set the translation of SMPL [3] as the chosen point. Next, we convert pose and shape parameters of a human to vertices, then we move the lowest vertex to the chosen point. After that, we convert the vertices to point cloud according to [1] and fit bounding box. Range image bridged point cloud generation (Sec 3.1) then generates a synthetic human that adhere to the view-dependent property of LiDAR point cloud. To filter out invalid insertion, we conduct following judgements:

- The occlusion rate of inserted synthetic human within the scene is less than 70%.
- The maximum Intersection over Union (IoU) between the bounding boxes of the inserted individual and those previously inserted should be less than 0.35.
- The occlusion rate of individuals previously inserted is less than 70%.

If all these judgements are satisfied, this insertion is valid, and we repeat the above insertion process until we achieve the wanted number of insertion for this frame. Otherwise, we consider this insertion as a failure, then we choose another distance and rerun above insertion process with the chosen distance. If we have 10 failures, the insertion for this frame is done.

Bi-directional tracking filter. We utilize AB3DMOT [4] as our tracker. For predicted bounding boxes with confidence score less than 0.5, we discard them before tracking. If we have unmatched tracking results, we discard them immediately instead of keeping them alive for a while. Tracklets with length less than 3 are discarded , while those with length longer than 3 but moving distance less than 2m are discarded as well. This is because HuCenLife and STCrowd dataset are both collected by located LiDAR (no traversals).

Key joints selection. We select six key joints representing the arms, legs, trunk, and head. Since the realistic occlusion in our synthetic data may make some body parts invisible, we filter out invisible parts based on the number of points within that part away from the closest key joint. Specifically, for each key joint, if there exists less than 10 points within the radius (0.4m, 0.22m, 0.3m, 0.15m for trunk, legs, head, arms, respectively), we filter out these parts.

Training setting. Our model is trained stage by stage. Our learning rate is 0.001 for stage 1 training, and 0.0001 for stage 2, stage 3, and finetune. We use AdamW as our optimizer. We train our model on 8 A40 GPUs for 12 hours each round. For fair comparison, we conduct no data augmentations on our model and baselines.

B. Data Diversity

Figure 1. Visualization of real data and synthetic data.

We improve the diversities of clothes and attachments by **synthesizing similar noise around the body** in data augmentation, as Fig. 1. In fact, due to the sparsity of LiDAR points, the noise caused by clothes **is not obvious**. To show the diversity of action, we Visualization of few synthetic human actions in Fig. 2.

Figure 2. Visualization of few synthetic human actions.

C. More result visualization

Fine-Grained Perception Enhancement is designed to ease the occlusion problem. E_7 in ablation study has shown its effectiveness for the whole data with **self- or external occlusions**. We show the improvement from E_6 to E_7 for occluded instances with Fig. 3 on HuCenLife test set.

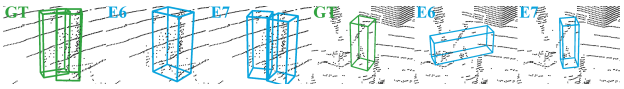

Figure 3. Visualization of few synthetic human actions.

References

[1] Peishan Cong, Xinge Zhu, and Yuexin Ma. Input-output balanced framework for long-tailed lidar semantic segmentation. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2021. 1

[2] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 1

[3] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Smpl: A skinned multi-person linear model. *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, 2015. 1

[4] Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. 3d multi-object tracking: A baseline and new evaluation metrics. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10359–10366. IEEE, 2020. 1