Appendix

A. More Dataset and Implementation Details

A.1. Dataset Details

Waymo Open dataset [?] collects 1,000 2D video sequences and 3D point clouds at synchronized time steps. There are around 200 frames captured at 10Hz in each video. Raw data from 5 LiDARs (1 mid-range and 4 short-range) and 5 cameras (1 front and 4 sides) is provided. In experiments, 2D images from the 'front' camera and 3D point clouds from the 'top' LiDAR are utilized. The temporal information from videos is not used. There are 3 object categories annotated with 2D bounding boxes in the dataset, i.e., 'vehicles', 'pedestrians', and 'cyclists'.

Cityscapes dataset [?] is a large-scale dataset for the autonomous driving scenario. The training and validation sets contain 2975 and 500 images, respectively. Each image is annotated with high quality pixel-level instance annotations. All images are of a resolution of 1024×2048 pixels. Feature transferability experiments are conducted on the instance segmentation task, which involves 8 object categories.

PASCAL VOC dataset [?] is consisted of the PASCAL VOC 2007 and 2012. Feature transferability experiments are conducted on the object detection task, which involves 20 object categories. Subsets trainval2007 and trainval2012 are used for training, and subset test2007 is used for validation. The training and validation sets contain 17k and 5k images, respectively.

A.2. Implementation Details

Training MoCo v2 on Waymo Open dataset [?] As for the MoCo feature learning for segment label initialization, we extend the MoCo v2 [**?**] with 3D point clouds as the additional inputs. Given the 3D segment ς_i, PointNet [**?**] is applied to extract a 1024-d 3D shape feature. Given the image x and the 2D bounding box b_i, ResNet-50 [**?**] is applied to extract a 2048-d image appearance feature from 2D image patch within bounding box b_i. We conduct the MoCo feature learning based on the concatenation of the 3D shape feature and 2D image appearance feature, which are further fed into a two-layer MLP with 128-d intermediate channels as in MoCo v2.

For the MoCo feature learning, augmentations for 2D image patches are exactly the same as in MoCo v2 [**?**]. Augmentations for 3D segments mainly follow PointNet [**?**], which consist of 1) random rotate along the up-axis with the angle in $[0, 2\pi]$, 2) horizontal random flip with probability of 0.5, and 3) random dropout points with the probability uniformly sampling from $[0, 0.875]$, and pad to 1024 points by re-sampling from the kept points.

During the MoCo feature learning, all hyperparameters follow the MoCo v2 [**?**]. After that, k-means clustering [**?**] is performed on the 3072-d features, which is the concatenation of the 1024-d 3D shape feature and 2048-d 2D image appearance feature.

Training Segment Labeling Networks on Waymo Open dataset [?] The segment labeling network N takes both the 2D image and 3D point cloud as input. Given the 3D segment ς_i, PointNet [**?**] is applied to extract a 1024-d 3D shape feature. Given the image x and the 2D bounding box b_i, Fast R-CNN [**?**] with ResNet-50 [**?**] is applied to extract a 1024-d 2D image appearance feature. The concatenation of these two features is further fed into a linear classifier. The Batch Normalization (BN) [**?**] layers are replaced with Synchronized BN [**?**]. Models are trained on images of shorter side $\{480, 512, 544, 576, 608, 640, 672, 704, 736, 768, 800\}$ pixels. In inference, images are resized so that the shorter side is 800 pixels.

In SGD training, 128 jittered segments (at a positive-negative ratio of 1:3) are sampled for each image. The networks are trained on 8 GPUs with 4 images per GPU for 6k iterations. The learning rate is initialized to 0.04 and is divided by 10 at the 3k and the 4k iterations. The weight decay and the momentum parameters are set as 10^{-4} and 0.9, respectively.

Training Object Detectors on Waymo Open dataset [?] As for the object detector, we utilize FPN [**?**] with ResNet-50 [**?**] as the backbone. The Batch Normalization (BN) [**?**] layers are replaced with Synchronized BN [**?**]. The other choice of hyperparameters for Faster R-CNN [**?**] follows the latest Detectron2 [**?**] code base, which is briefly presented here. Models are trained on images of shorter side $\{480, 512, 544, 576, 608, 640, 672, 704, 736, 768, 800\}$ pixels. In inference, images are resized so that the shorter side is 800 pixels. Anchors are of 5 scales and 3 aspect ratios. 1k region proposals are generated at an NMS threshold of 0.7. During inference, detection results are derived at an NMS threshold of 0.3.

In SGD training, 256 anchor boxes (at a positive-negative ratio of 1:1) and 128 region proposals (at a positive-negative ratio of 1:3), are sampled for RPN and Fast R-CNN, respectively. The networks are trained on 8 GPUs with 4 images per GPU for 6k iterations. The learning rate is initialized to 0.04 and is divided by 10 at the 3k and the 4k iterations. The weight decay and the momentum parameters are set as 10^{-4} and 0.9, respectively.

Transfer learning on Cityscapes dataset [?] and PASCAL VOC dataset [?] Mask R-CNN [**?**] and Faster R-CNN [**?**] are utilized for instance segmentation and object detection, respectively. ResNet-50 [**?**] with FPN [**?**] is utilized as the backbone. For all experiments, the Batch Normalization (BN) [**?**] layers are replaced with Synchronized

foreground threshold η	vehicle			pedestrains			cyclist		
	iter1	iter2	iter3	iter1	iter2	iter3	iter1	iter2	iter3
0.999	23.0	24.9	26.2	11.5	12.2	15.0	0.0	1.0	0.0
0.99	26.3	26.9	27.6	17.1	21.8	23.1	2.3	2.3	1.7
0.95	26.9	27.7	28.5	21.1	23.8	25.6	5.2	5.5	6.0
0.90	27.7	27.7	27.8	21.8	24.6	26.1	3.6	4.5	5.9
0.80	27.0	27.7	27.8	22.6	25.5	26.5	4.5	4.3	3.8

Table 1: Ablation on foreground threshold η in the iterative segment labeling on Waymo Open validation. The default setting is highlighted.

BN [**?**]. The other choice of hyperparameters mainly follow [**?**], which is briefly presented here.

For experiments on Cityscapes, models are trained on images of shorter sides {800, 832, 864, 896, 928, 960, 992, 1024} pixels, and tested on its original resolution of 1024×2048 pixels. In SGD training, the networks are trained on 8 GPUs with 1 image per GPU for 24k iterations. The learning rate is initialized to 0.01 and is divided by 10 at the 18k iterations.

For experiments on PASCAL VOC, models are trained on images of shorter sides {480, 512, 544, 576, 608, 672, 704, 736, 768, 800} pixels, and tested on images with a shorter side of 800 pixels. In SGD training, the networks are trained on 8 GPUs with 2 images per GPU for 54k iterations. The learning rate is initialized to 0.02 and is divided by 10 at the 36k and 48k iterations.

B. More Ablations

Foreground Probability Threshold for Iterative Labeling During the process of iterative segment labeling, a segment is labeled as background if its estimated foreground probabilities are less than η for all clusters. As shown in Table 1, our iterative segment labeling is robust to the choice of foreground threshold within a wide range.

Visualization of Learned Feature Space Since our method is unsupervised and has no access to semantic labels during training, it is interesting to see how well the learned features align with the semantics. Here, we show the t-SNE [**?**] for features learned by the segment labeling network after 10 rounds iterative labeling, which is the concatenation of the 1024-d 3D shape feature and the 1024-d 2D image appearance feature. Only candidate segments labeled as foreground by our method are shown in the embedding. Figure 1 and Figure 2 show the t-SNE visualizations, where each embedding point is represented by an image patch and a LiDAR segment, respectively. The learned features could distinguish objects of different categories (e.g., car, pedestrian, and traffic cones) as well as the semantic subgroups of the same category (e.g., front, side, and rare views).

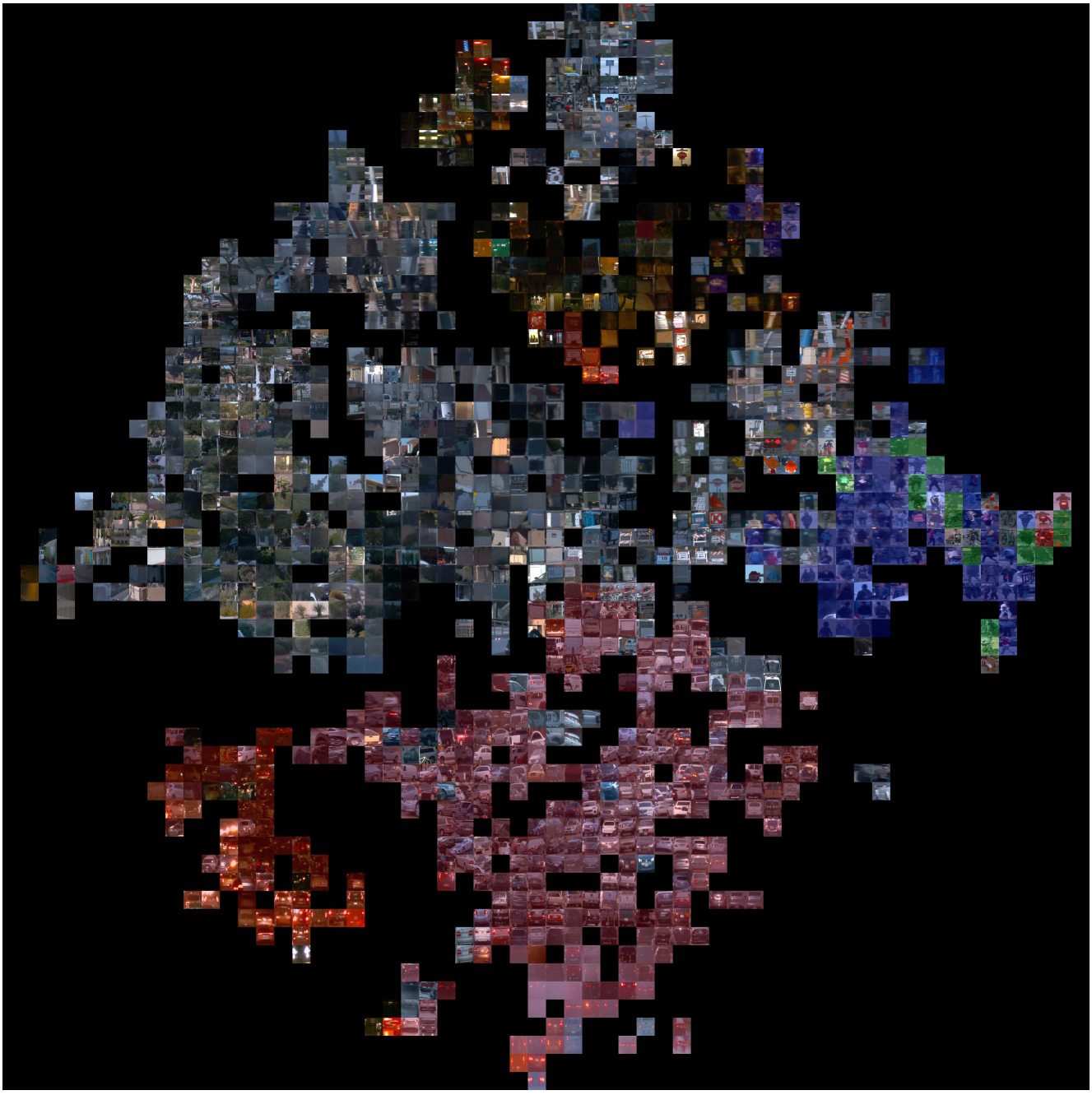

Figure 1: Visualization of t-SNE for features from the segment labeling network. Each embedding point in t-SNE corresponds to an candidate segment labeled as foreground by our method. Randomly sampled embedding points are visualized by the corresponding 2D image patches. Different color blendings represent difference categories: red for vehicle, blue for pedestrian, green for cyclist, and the remainings are unlabeled in the Waymo Open dataset.

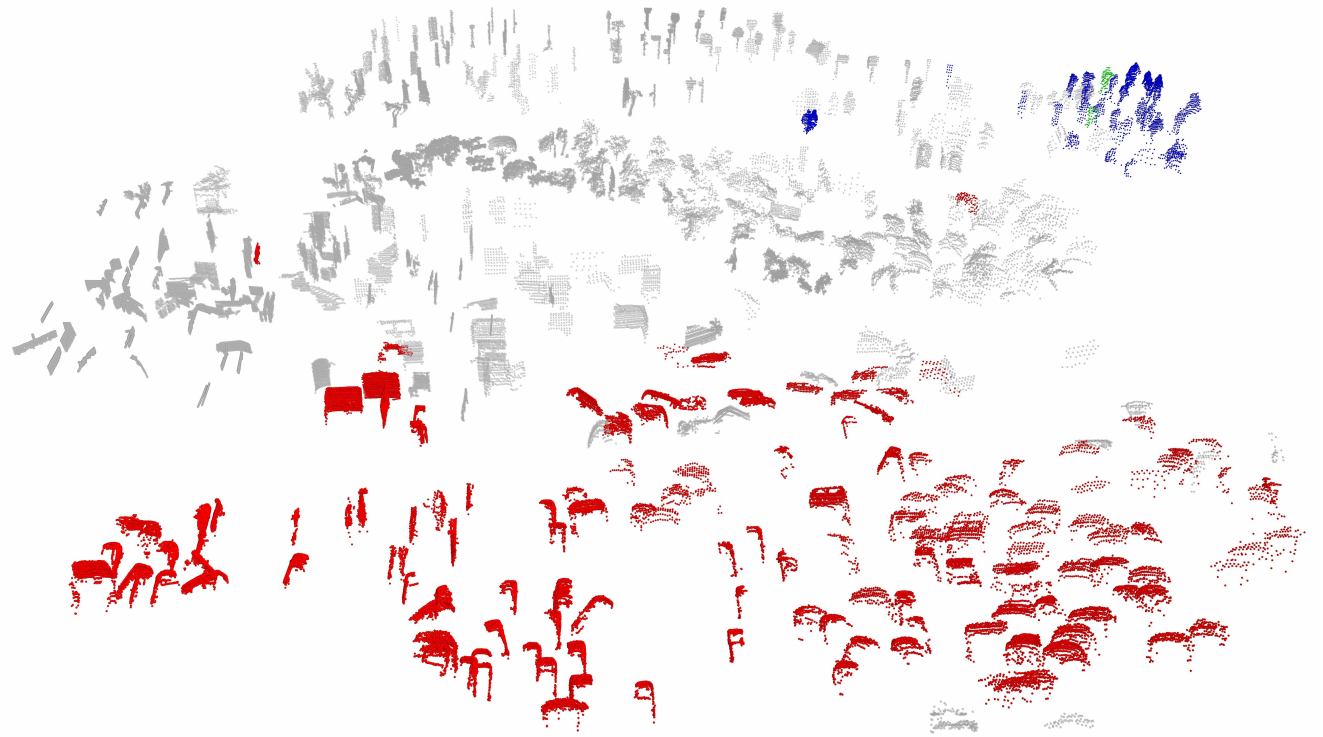

Figure 2: Visualization of t-SNE for features from the segment labeling network. Each embedding point in t-SNE corresponds to an candidate segment labeled as foreground by our method. Randomly sampled embedding points are visualized by the corresponding 3D LiDAR segments. Different colors represent difference categories: red for vehicle, blue for pedestrian, green for cyclist, and the remainings are unlabeled in Waymo Open dataset.