Transferable Semi-Supervised 3D Object Detection From RGB-D Data - Supplementary Materials

Yew Siang Tang Gim Hee Lee
Department of Computer Science, National University of Singapore
{yewsiang.tang, gimhee.lee}@comp.nus.edu.sg

In Sec. A, we discuss the costs of different types of labels. In Sec. B, we observe the performance with varying amounts of 3D box labels for C_{2D} classes. In Sec. C and Sec. D, we elaborate on the details of the networks and the details of the training procedures, respectively. In Sec. E, we provide additional qualitative results and figures on the SUN-RGBD dataset and in Sec. F, qualitative results for the KITTI dataset.

A. Time Costs of Labels

The SUN-RGBD dataset [25] required 2,051 hours to label 64,595 3D bboxes, which is an average of 114s per object. In contrast, the average time to label a 2D bbox according to [17] is 35s per object but can be as fast as 7s when using their proposed labeling method with no loss of accuracy. Hence, it is potentially 3-16 times faster to label 2D compared to 3D bboxes.

B. In-Category Semi-supervision Performance

In the main paper, we assumed that there are no 3D box labels for *weak* classes C_{2D}, which is a cross-category semi-supervised setting. In Fig. 1, we train our model and the baseline "FPN*" on varying amounts of 3D box labels for C_{2D} to understand the performance of our model in an in-category semi-supervised setting for the SUN-RGBD dataset. When the percentage of 3D box labels used for C_{2D} is 100%, the semi-supervised baseline "FPN*" (green line) becomes the fully-supervised "FPN*" (51.5% mAP) and our proposed semi-supervised method (blue line) becomes the fully-supervised "FPN* + BoxPC Refine" (53.2% mAP).

We observe that our proposed method always performs better than the baseline for different percentages of 3D box labels of C_{2D} classes. This demonstrates the usefulness of our method even when labels are available for C_{2D} classes. Additionally, we note that the baseline with 50% 3D box labels available for C_{2D} achieves a similar performance to our proposed method with 0% 3D box labels for C_{2D}, which demonstrates the effectiveness of the knowledge that has been transferred from C_{3D} classes.

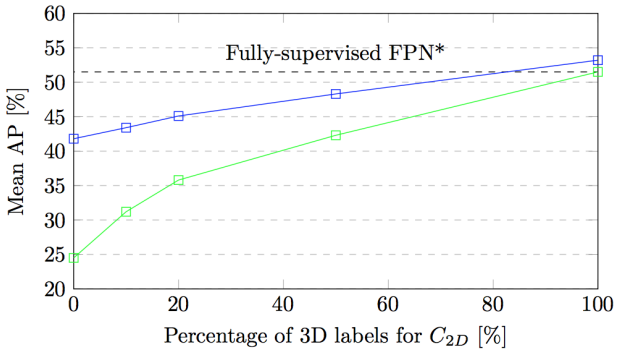


Figure 1. 3D object detection mAP on SUN-RGBD *val* set of our proposed model with different percentages of 3D labels for C_{2D}. The blue and green lines correspond to the "Ours + BoxPC + R + P" and "FPN*" semi-supervised settings, respectively.

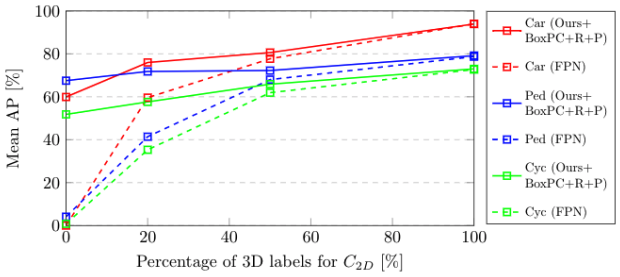


Figure 2. 3D object detection mAP on KITTI *val* set of our proposed model with different percentages of 3D labels for C_{2D} (3D IoU threshold of 0.25).

Similarly, we perform the same in-category semi-supervised setting for the KITTI dataset by varying the amount of 3D box labels for C_{2D}. In Fig. 2, our proposed model (solid) trained under the in-category semi-supervised setting still constantly performs better than the baseline (dashed) for all percentages of 3D labels.

C. Network Details

C.1 Network Architecture of Baselines

Fig. 3 shows the network architecture for the original Frustum PointNets (FPN) [2]. The same architecture is used

to obtain the results for "FPN" and the baseline "FPN*" in Tab. 1 of the main paper. We remove the one hot class vectors given as features to the f_{seg} and f_{box} networks to get the stronger baseline "FPN* w/o OneHot". The performance improves because in the cross-category semi-supervised setting, the network does not train on the strong labels of inference classes. Hence, having class information does not help the network during inference.

C.2 Network Component Details

In this section, we describe the details of the network components in the baseline and proposed models for the cross-category semi-supervised 3D object detection (CS3D) setting. The fully-supervised 3D object detection (FS3D) setting uses the same components except for minor changes.

The network components are given in Fig. 4. We use the v1 instance segmentation and v1 box estimation networks in FPN [2] as the instance segmentation and box estimation networks in our baseline and proposed models. The BoxPC Fit networks also use PointNet [3] and MLP layers to learn the BoxPC features.

In the CS3D setting, we remove the one hot class vector that is originally concatenated with the "Global Features" of the v1 instance segmentation network because we are performing class-agnostic instance segmentations. In the FS3D setting, the one hot class vector is added back.

The v1 box estimation network is composed of the T-Net and Box Estimation PointNet. The T-Net gives an initial prediction for the center of the box b_x, b_y, b_z which is used to translate the input point cloud to reduce translational variance. Then, the Box Estimation PointNet makes the box predictions using the translated point cloud. Both CS3D and FS3D settings use the same box estimation networks. The details of the loss functions to train the box estimation networks can be found in [2].

In the CS3D setting, the BoxPC Fit network learns class-agnostic BoxPC Fit between 3D boxes and point clouds. In the FS3D setting, we concatenate a one hot class vector to the "BoxPC Features" to allow the network to learn BoxPC Fit that is specific to each class.

D. Training Details

D.1 Training of BoxPC Fit Network

As discussed in the paper, we have to sample from 2 sets of perturbations $\mathbf{P}^+$ and $\mathbf{P}^-$ to train the BoxPC Fit Network to understand what is a good BoxPC fit. To sample perturbations $\delta = [\delta_x, \delta_y, \delta_z, \delta_h, \delta_w, \delta_l, \delta_\theta]$ from either $\mathbf{P}^+$ or $\mathbf{P}^-$, we uniformly sample center perturbations $\delta_x, \delta_y, \delta_z \in [-0.8, 0.8]$, size perturbations $\delta_h, \delta_w, \delta_l \in [-0.2, 0.2]$ and rotation perturbations $\delta_\theta \in [0, \pi]$. We perturb a 3D box label B^* to obtain a perturbed 3D box label $B^* - \delta$ with a sampled perturbation δ. Next, we check if $B^* - \delta$ has an IOU with B^* that is within the range specified by the set $\mathbf{P}^+$ or $\mathbf{P}^-$, i.e if $\alpha^+ \leq IOU(B^* - \delta, B^*) \leq \beta^+$ or $\alpha^- \leq IOU(B^* - \delta, B^*) \leq \beta^-$, respectively. We accept and use it as a single input if it satisfies the IOU range for the set. We repeat the process until we have enough samples for a minibatch. Specifically, each minibatch has equal number of $B^* - \delta$ samples from $\mathbf{P}^+$ and $\mathbf{P}^-$.

D.2 Cross-category Semi-supervised Learning

Let C_A = {`bathtub, bed, toilet, chair, desk`} and C_B = {`dresser, nightstand, sofa, table, bookshelf`}. We train with $C_{3D} = C_B$ and $C_{2D} = C_A$, i.e. we train on the 2D and 3D box labels of C_B and the 2D box labels of C_A, to obtain the evaluation results for CS3D on C_A. We train with $C_{3D} = C_A$ and $C_{2D} = C_B$ to get the evaluation results for CS3D on C_B.

SUN-RGBD For our 2D object detector, we train a Faster RCNN [1,4] network on all classes $C = C_{2D} \cup C_{3D}$. When we train the BoxPC Fit network on classes C_{3D}, we set the perturbation parameters to $\alpha^+ = 0.7, \beta^+ = 1.0, \alpha^- = 0.01, \beta^- = 0.25$. The loss weights for the BoxPC Fit network are set to $w_{cls} = 1, w_{reg} = 4$. When we train the 3D object detector, we set the lower and upper bound boxes for the relaxed reprojection loss to $B^*_{lower} = B^*_{2D}, B^*_{upper} = 1.5\ B^*_{2D}$ and volume threshold for all classes to $V = 0$. The loss weights for the 3D object detector are set to $w_{c1-reg} = 0.1, w_{c2-reg} = 0.1, w_{r-cls} = 0.1, w_{r-reg} = 2, w_{s-cls} = 0.1, w_{s-reg} = 2, w_{corner} = 0.1, w_{fit} = 0.05, w_{reproj} = 0.0005, w_{vol} = 0, w_{s-var} = 0.1$.

KITTI For our 2D object detector, we use the released detections from FPN [2] for fair comparisons with FPN. We set the perturbation parameters to $\alpha^+ = 0.8, \beta^+ = 1.0, \alpha^- = 0.01, \beta^- = 0.4$ when we train the BoxPC Fit network on classes C_{3D}. The loss weights of the BoxPC Fit network are set to $w_{cls} = 1, w_{reg} = 4$. We set the lower and upper bound boxes for the relaxed reprojection loss to $B^*_{lower} = B^*_{2D}, B^*_{upper} = 1.5\ B^*_{2D}$ for *Pedestrians* and *Cyclists*, and $B^*_{lower} = B^*_{2D}, B^*_{upper} = 1.25\ B^*_{2D}$ for *Cars* when we train the 3D object detector on the classes C. We set the volume threshold for *Cars* to $V^{car} = 10$ and 0 for the other classes. The loss weights for the 3D object detector are set to $w_{c1-reg} = 1, w_{c2-reg} = 1, w_{r-cls} = 1, w_{r-reg} = 20, w_{s-cls} = 1, w_{s-reg} = 20, w_{corner} = 10, w_{fit} = 1, w_{reproj} = 0.2, w_{vol} = 1, w_{s-var} = 2$.

D.3 Fully-supervised Learning

In the fully-supervised setting, we train and evaluate on C.

SUN-RGBD We use the same 2D detector as in the CS3D setting. When we train our BoxPC Fit network, we set the perturbation parameters to $\alpha^+ = 0.6, \beta^+ = 1.0, \alpha^- = 0.05, \beta^- = 0.4$. The loss weights of the BoxPC Fit network are set to $w_{cls} = 1, w_{reg} = 4$. For our 3D object detector, we set the parameters to $w_{c1-reg} = 0.1, w_{c2-reg} =$

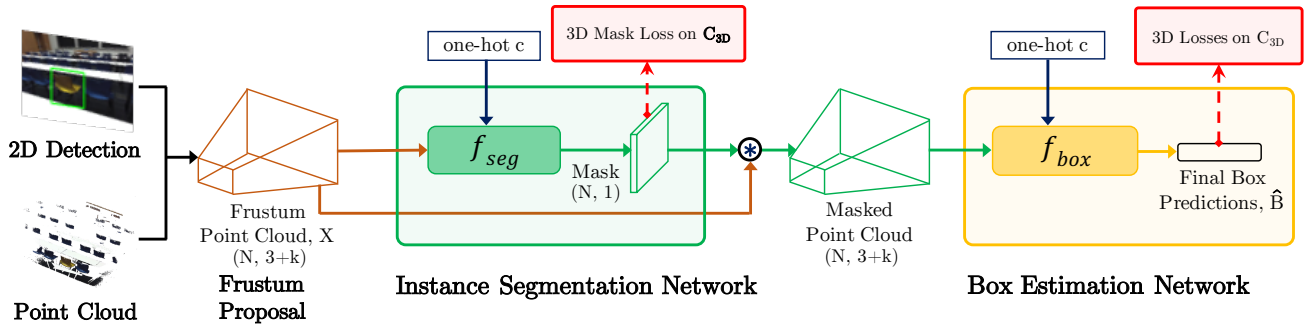
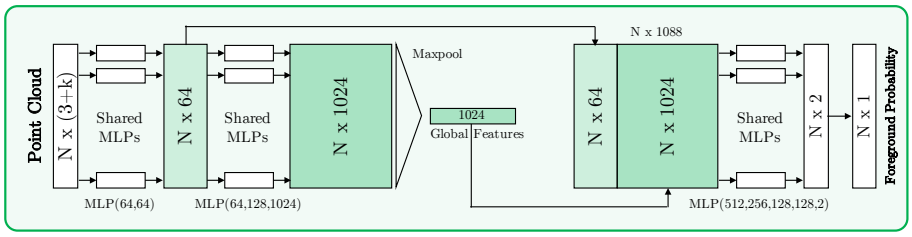
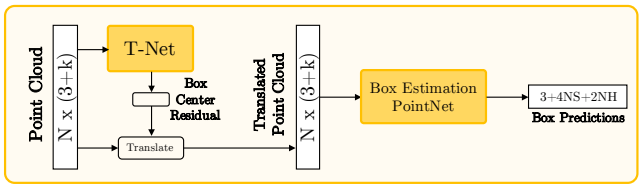


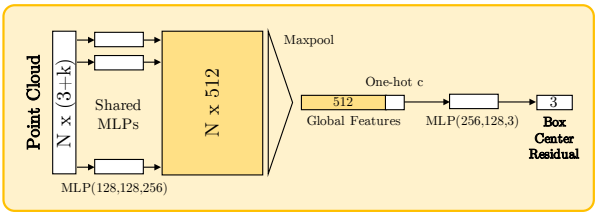
Figure 3. Network architecture for the original Frustum PointNets for fully-supervised 3D object detection. This network corresponds to "FPN" and the baseline "FPN*" in Tab. 1 of the main paper. To obtain the stronger baseline "FPN* w/o OneHot", we remove the one hot class vectors that are given as features to f_{seg} and f_{box}.



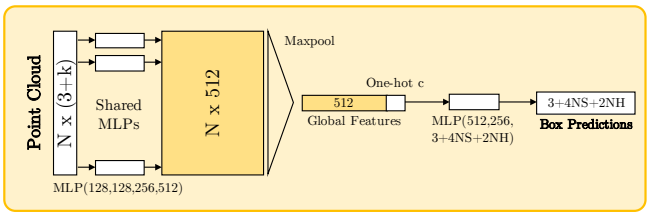
Instance Segmentation Network



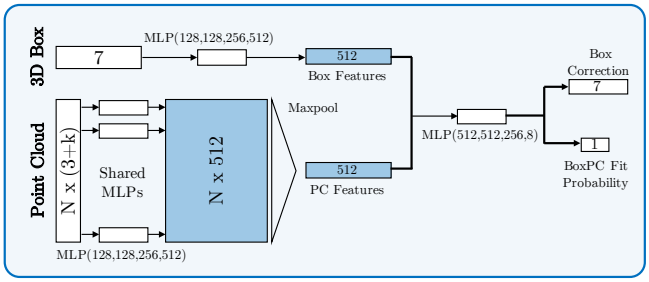
Box Estimation Network



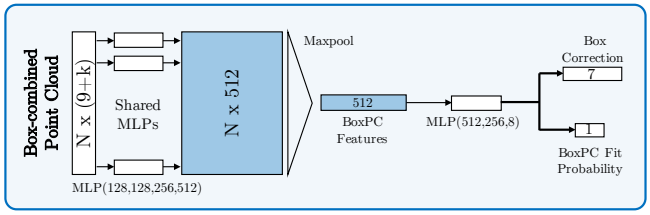
T-Net



Box Estimation PointNet



BoxPC Fit Network (Independent BoxPC Features)



BoxPC Fit Network (Combined BoxPC Features)

Figure 4. Network component details for our baseline and proposed models that are used in cross-category semi-supervised 3D object detection. In fully-supervised setting, a one hot class vector is added to the "Global Features" of the Instance Segmentation Network and the "BoxPC Features" of the BoxPC Fit Network.

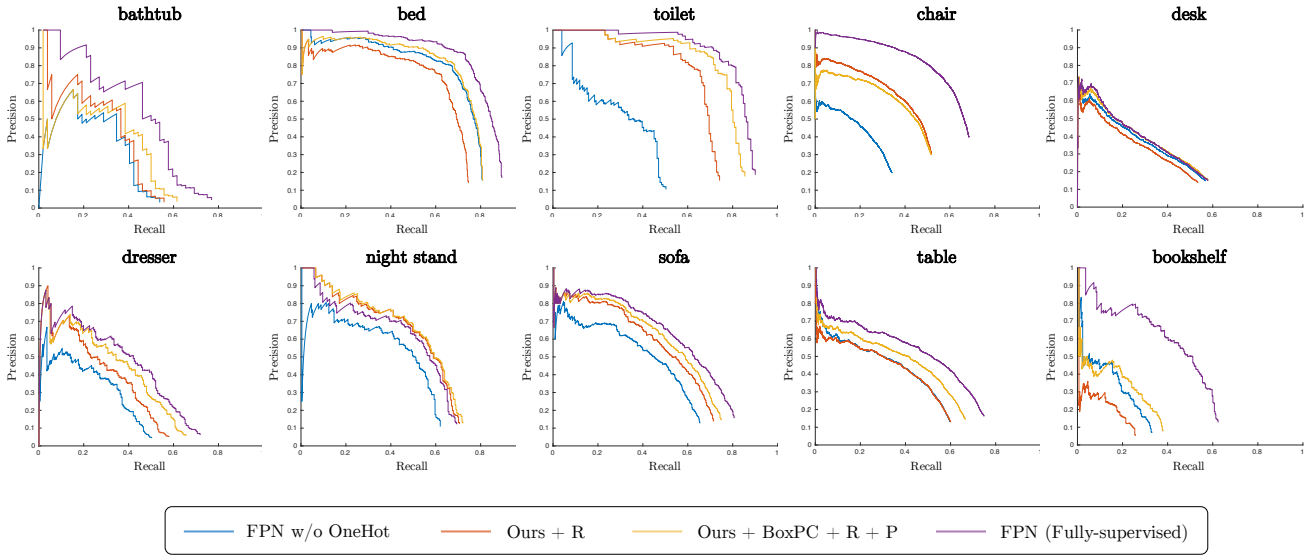


Figure 5. Precision recall (PR) curves for 3D object detection on SUN-RGBD *val* set with different methods. "FPN (Fully-supervised)" is the fully-supervised model that gives an upper-bound performance for the rest of the models which are cross-category semi-supervised.

$0.1, w_{r-cls} = 0.1, w_{r-reg} = 2, w_{s-cls} = 0.1, w_{s-reg} = 2, w_{corner} = 0.1$. Next, we use the trained BoxPC Fit network to refine the 3D box predictions of this fully-supervised model without further training.

KITTI We use the same 2D detections as the CS3D setting. We train one BoxPC Fit network for each of the 3 classes to allow each network to specialize on improving box predictions for a single class. For *Cars*, we set $\alpha^+ = 0.7, \beta^+ = 1.0, \alpha^- = 0.6, \beta^- = 0.8$. For *Pedestrians*, we set $\alpha^+ = 0.7, \beta^+ = 1.0, \alpha^- = 0.3, \beta^- = 0.7$. For *Cyclists*, we set $\alpha^+ = 0.7, \beta^+ = 1.0, \alpha^- = 0.5, \beta^- = 0.7$. The loss weights of the BoxPC Fit network are set to $w_{cls} = 0, w_{reg} = 4$. Next, we use the trained BoxPC Fit networks to refine the 3D box predictions of the 3D object detector model released by [2] without further training.

E. Additional Results for SUN-RGBD

In Fig. 5, we plot the precision-recall curves for different methods to study the importance of each proposed component. All the methods are cross-category semi-supervised except for "FPN (Fully-supervised)", which is the fully-supervised FPN [2] that gives an upper-bound performance for the methods. We observe that "Ours + BoxPC + R + P" (in yellow) has higher precision at every recall than the baseline "FPN w/o OneHot" (in blue) for almost all classes.

We also provide additional qualitative results on the SUN-RGBD dataset in Fig. 6. The predictions made by the baseline model tend to be large and inaccurate due to the lack of strong labels in the *weak* classes. In the third example of Fig. 6, we see that the predictions by the baseline model can also be highly unnatural as it cuts through the wall. On the contrary, we observe that the proposed model's predictions tend to be more reasonable and closer to the fully-supervised model despite not having strong labels for the *weak* classes. In the last two examples of Fig. 6, the heavy occlusions in the scene makes it difficult for the baseline and proposed models to make good predictions.

F. Qualitative Results for KITTI

In Fig. 7 and Fig. 8, we provide qualitative comparisons between the baseline, fully-supervised and proposed models for the KITTI dataset. In both Fig. 7 and Fig. 8, we again observe that the proposed model is closer to the fully-supervised model than the baseline model. Notably, in the first example (top left) and ninth example (bottom right) of Fig. 7, we observe that the proposed model is able to make significantly good predictions on *Pedestrians* despite the crowded scene.

For almost all of the *Car* instances, the baseline model makes excessively small predictions because it was trained on *Pedestrian* and *Cyclist* classes which are much smaller. Our proposed model is able to make better predictions on *Cars* but the orientation of the 3D box predictions can be improved further.

In the last two examples of Fig. 8, we show some of the difficult scenes for our baseline and proposed models. This is because there are heavy occlusions and poor understanding of *Cars* since the models were trained on *Pedestrians* and *Cyclists*, which have very different shapes and sizes from *Cars*.

References

[1] Xinlei Chen and Abhinav Gupta. An implementation of faster rcnn with study for region sampling. *arXiv preprint arXiv:1702.02138*, 2017.

[2] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 918–927, 2018.

[3] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 1(2):4, 2017.

[4] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

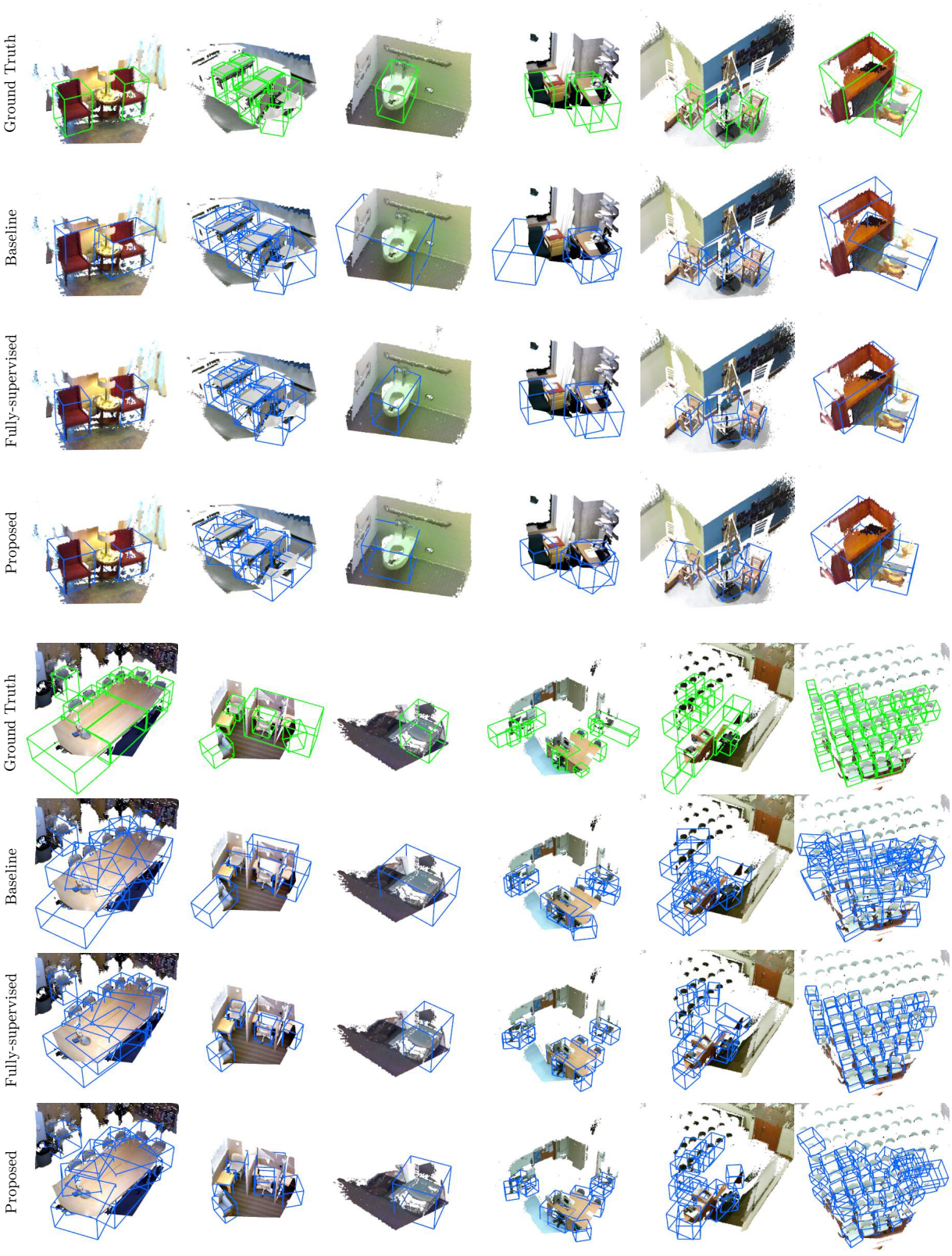


Figure 6. Additional qualitative comparisons between the baseline, fully-supervised and proposed models on the SUN-RGBD *val* set. The baseline model's predictions tend to be large and inaccurate because it is unable to understand how to fit objects from the *weak* classes. The last two examples (bottom right) are difficult scenes for the baseline and proposed models due to heavy occlusions.

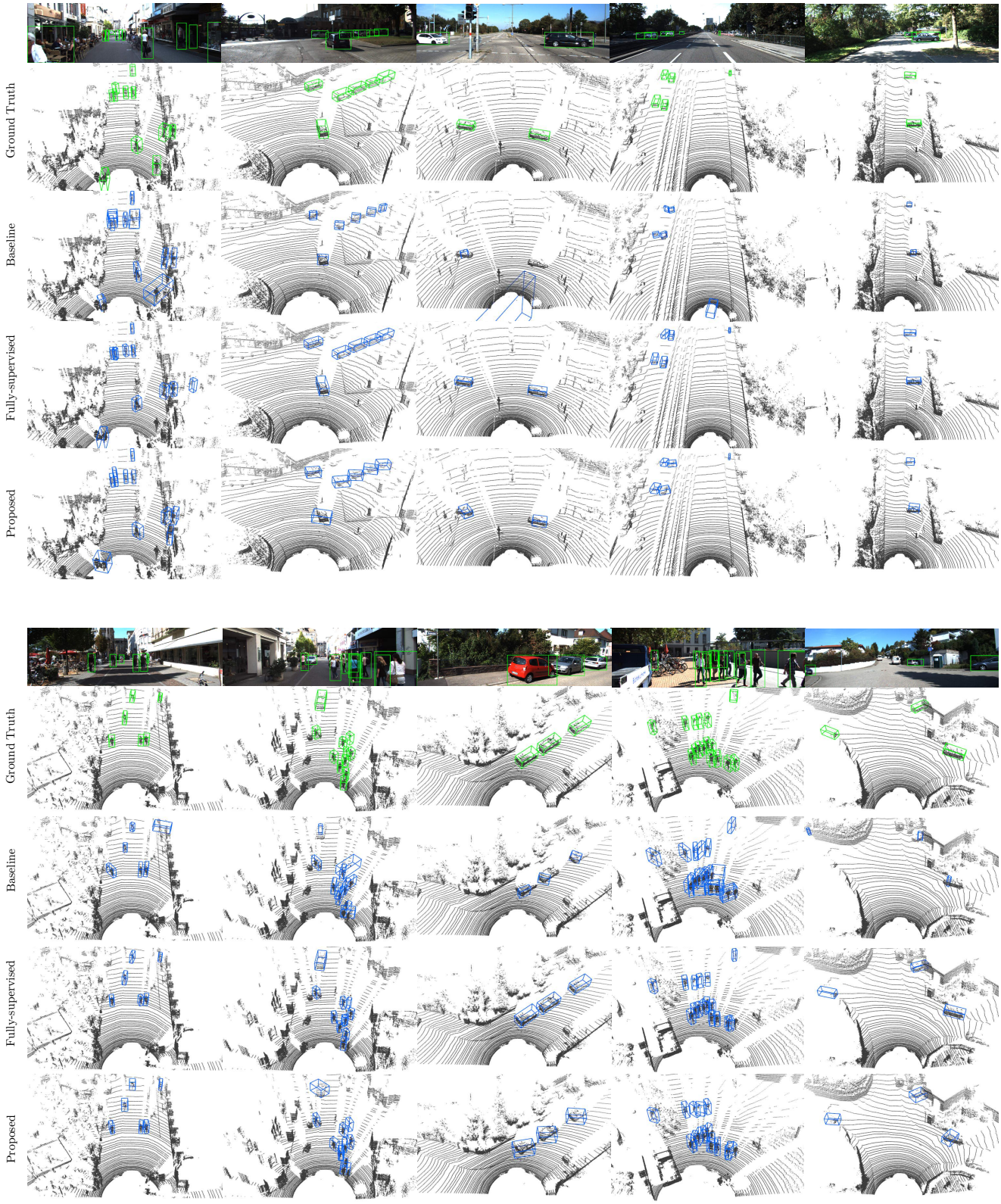


Figure 7. Qualitative comparisons between the baseline, fully-supervised and proposed models on the KITTI *val* set. The proposed model is much closer to the fully-supervised model than the baseline model. The baseline model's predictions for *Cars* tend to be inaccurate or excessively small due to the lack of a prior understanding on the scale of *Cars*.

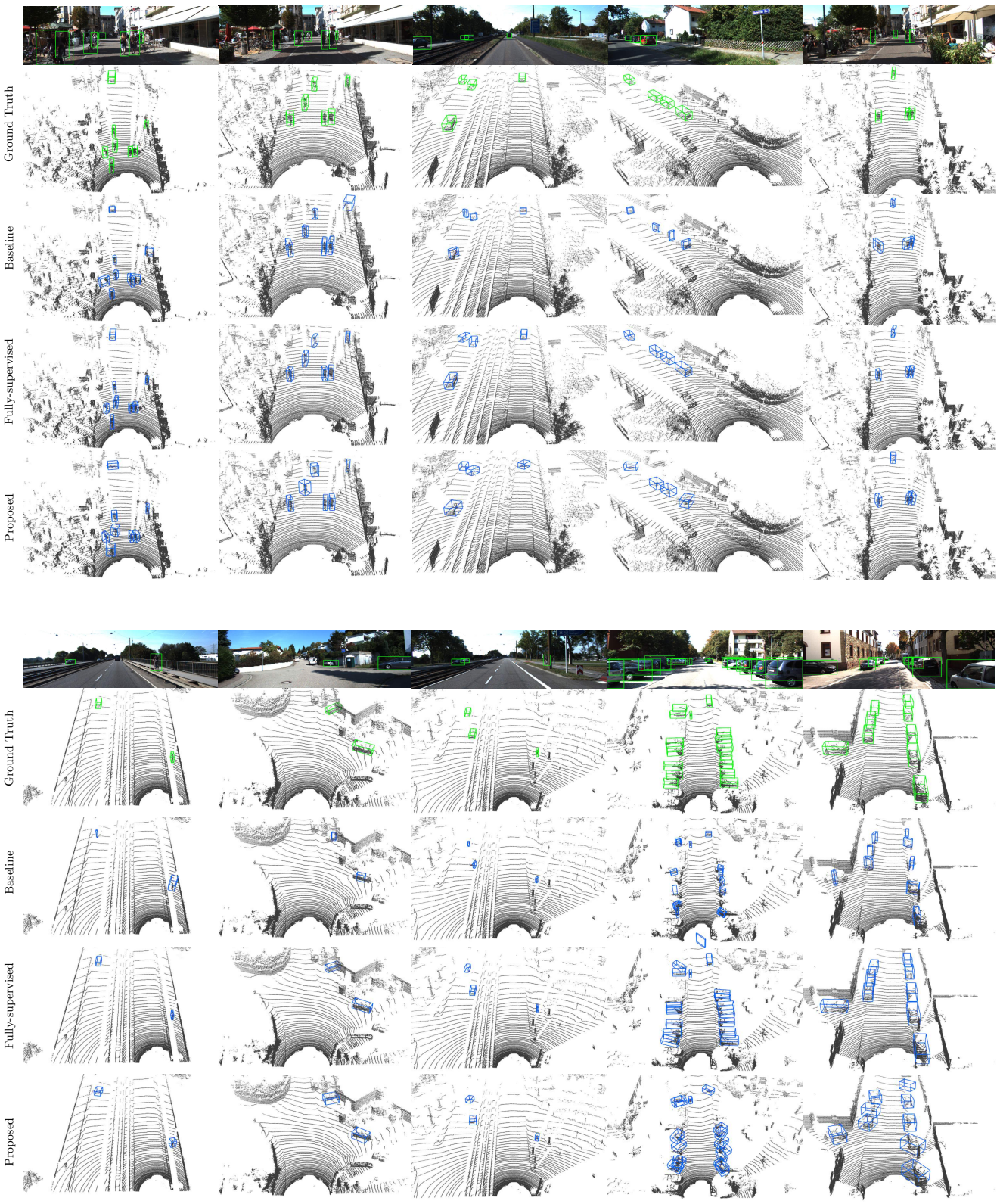


Figure 8. Additional qualitative comparisons between the baseline, fully-supervised and proposed models on the KITTI *val* set. The last two examples (bottom right) are difficult scenes for the baseline and proposed models. This is due to heavy occlusions and poor understanding on how to fit *Cars* using transferred 3D information from *Pedestrians* and *Cyclists*, which have very different shapes from *Cars*.