# Supplementary Material for ViT-WSS3D

Table 1. The results on the Waymo dataset.

| Method | Vehicle | Ped. | Cyc. |
|---|---|---|---|
| 100% Full label | 67.8 | 67.6 | 56.0 |
| 10% Full label | 58.1 | 53.2 | 19.8 |
| 10% F + 90% W (ours) | 67.2 | 67.5 | 54.2 |

## 1. Overview

The supplementary material provides some more specific information as follows:

- More implementation details, including dataset parameters, data augmentation, training configuration.
- More quantitative and qualitative results.

## 2. More implementation details

In this section, we will describe the extra settings of our method on KITTI [1] and SUN RGB-D [7] datasets. Note that all these settings are only for the teacher, and we do not change any setting of students.

**KITTI.** Following the previous methods [6, 2, 9], we filter the points outside the point cloud range, which is set as $x \in [0, 70.4], y \in [-40, 40], z \in [-3, 1]$. We let the detection head regress the object dimension by predicting the residual w.r.t. the average dimension of each class in the whole training split, where the average dimension is $d_x = 3.9, d_y = 1.6, d_z = 1.56$ for Car, $d_x = 0.8, d_y = 0.6, d_z = 1.73$ for Pedestrian, and $d_x = 1.76, d_y = 0.6, d_z = 1.73$ for Cyclist.

We leverage some commonly used data augmentations to learn a more powerful teacher. In order to increase the diversity of samples, we use the GT-Sampling [9] to complement the ground truths in each scene, where we sample 15, 8, and 8 samples for Car, Pedestrian, and Cyclist in each scene, respectively. Moreover, we use the global rotation and scale transformation with rotation range as $[-\frac{\pi}{4}, \frac{\pi}{4}]$ and scale ratio range as $[0.95, 1.05]$. We also use the random local rotation and translation with translation standard deviation as $std_x = 1.0, std_y = 1.0, std_z = 0.5$ and rotation range as $[-\frac{\pi}{4}, \frac{\pi}{4}]$. We sample points to form a fixed number (*i.e.*, 16384) input points and shuffle the points randomly to force the teacher invariant to input permutation [5].

For training the teacher, we use the AdamW [3] optimizer with $\beta_1 = 0.95, \beta_2 = 0.85$. We adopt the one-cycle learning rate and momentum schedule with max learning rate 0.0001, weight decay 0.01, and momentum 0.85 to 0.95. We train the teacher for 80 epochs with batches per GPU as 4.

**SUN RGB-D.** Following [8] and because the scale of the indoor scene is not as large as the outdoor scene, we do not filter points. Additionally, due to the high diversity of indoor objects, we do not use the average object dimensions, instead, we directly predict the 3D dimensions of objects.

For data augmentations, we first randomly flip the scenes horizontally with probability 0.5, then apply the global rotation and scale transformation with rotation range $[-\frac{\pi}{6}, \frac{\pi}{6}]$ and scale ratio range as $[0.85, 1.15]$. Finally, we sample 20000 points from the original point cloud for acceleration.

When training the teacher, we adopt the AdamW optimizer with $\beta_1 = 0.9, \beta_2 = 0.999$. We use the step learning rate schedule with max learning rate 0.0001, weight decay 0.0005, and we reduce the learning rate by $10\times$ at the 24 and 32 epoch. We train the teacher for 36 epochs with batches per GPU as 4.

## 3. Additional results

### 3.1. Results on Waymo

We report the performance (L1 mAP, PointPillars as the student) on the large-scale dataset, Waymo. Note that we choose the original 20% data as the 100% set, due to the limited computation resource. As shown in Tab. 1, compared with 100% full labels, our method reports similar performance while significantly reducing the annotation cost.

### 3.2. Quantative results

For a closer look at the differences between 3DIoUMatch [8] and our method, we pose the students performance for each category, shown in Tab. 2.

We find that under any data setting and any category, our method always guides the student to achieve significantly higher mAP, especially on those categories that cannot be well predicted by the original student (*e.g.*, Dresser and Bookshelf). In stark contrast, 3DIoUMatch performs worse than ours on these categories, sometimes hardly helping or

Table 2. The comparison results of 3DIoUMatch and our method on SUN RGB-D *val* split. We take the VoteNet as the student and report mAP@0.25.

| Method | Setting | Bed | Table | Sofa | Chair | Toilet | Desk | Dresser | Night stand | Bookshelf | Bathtub | Overall |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| VoteNet [4] | 100% Full | 84.5 | 49.6 | 68.3 | 78.0 | 90.2 | 25.3 | 29.2 | 62.3 | 35.4 | 75.1 | 59.8 |
| VoteNet [4] | 5% Full | 74.0 | 32.6 | 43.6 | 59.6 | 66.3 | 9.1 | 2.0 | 38.1 | 2.2 | 37.8 | 36.5 |
| 3DIoUMatch [8] | 5% Semi | 77.9 | 37.1 | 41.6 | 61.7 | 77.3 | 6.2 | 1.6 | 36.1 | 0.4 | 59.6 | 40.0 |
| Ours | 5% Full + 95% Weak | 82.8 | 42.7 | 59.6 | 73.8 | 71.5 | 22.0 | 25.0 | 57.7 | 12.2 | 76.4 | 52.4 |
| VoteNet [4] | 10% Full | 77.1 | 35.4 | 48.2 | 63.0 | 73.5 | 9.3 | 7.4 | 45.0 | 3.1 | 45.4 | 40.7 |
| 3DIoUMatch [8] | 10% Semi | 80.1 | 40.5 | 53.8 | 66.3 | 78.5 | 10.2 | 6.8 | 47.0 | 8.3 | 58.2 | 45.0 |
| Ours | 10% Full + 90% Weak | 84.6 | 44.6 | 63.3 | 74.4 | 88.1 | 22.2 | 26.6 | 63.4 | 21.3 | 81.7 | 57.0 |
| VoteNet [4] | 20% Full | 80.0 | 43.2 | 57.9 | 70.1 | 78.7 | 14.6 | 13.0 | 50.0 | 12.7 | 53.2 | 47.4 |
| 3DIoUMatch [8] | 20% Semi | 80.7 | 43.8 | 56.9 | 69.5 | 81.5 | 13.8 | 14.8 | 46.8 | 17.4 | 62.9 | 48.8 |
| Ours | 20% Full + 80% Weak | 85.9 | 48.8 | 65.1 | 73.2 | 89.5 | 27.2 | 26.9 | 63.7 | 29.4 | 78.0 | 58.8 |

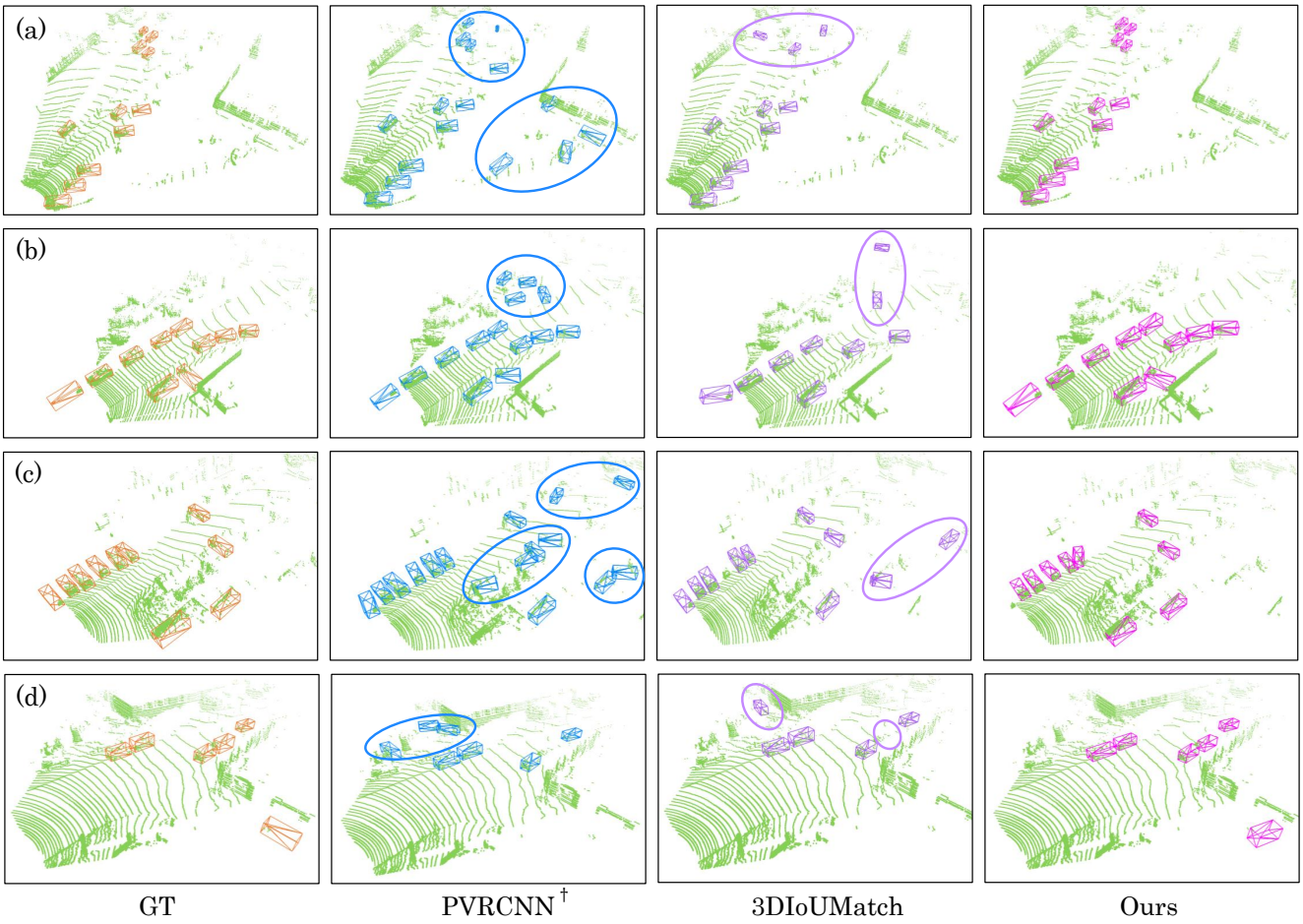


Figure 1. The visualization of pseudo labels from different methods on KITTI with 2% full data. † means training detectors on fully-labeled data and then use them to infer pseudo boxes.

even hindering the student. For example, under the 5% full data setting, the student performs worse than the original student on Desk, Dresser, and Bookshelf under its guidance.

We argue that this is due to the dependence between the teacher and the student brought by the teacher-student mutual learning framework, where the performance of students can influence the performance of the teacher. Unlike 3DIoUMatch, our method makes the teacher independent

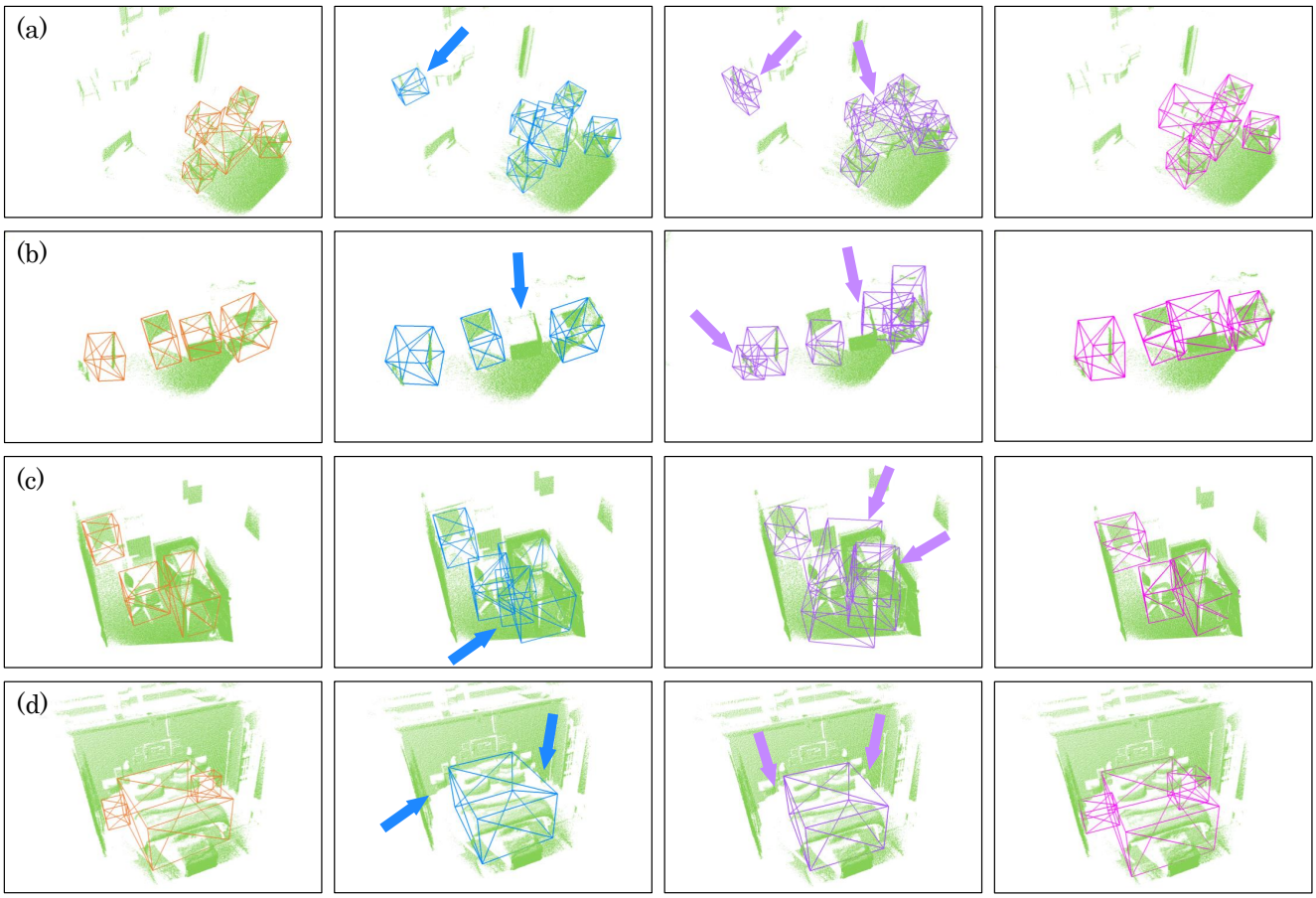


Figure 2. The visualization of pseudo labels from different methods on SUN RGB-D with 5% full data. † means training detectors on fully-labeled data and then use them to infer pseudo boxes.

of students, and the results have shown the transcendence of our method all around.

### 3.3. Qualitive results

We further show some additional visualizations of pseudo labels from different methods in Fig. 1 and Fig. 2. We mark the obvious errors with circles and arrows. Our method can generate pseudo labels almost identical to GTs, while 3DIoUMatch generates many false positives and false negatives. The visualization demonstrates the much better quality of pseudo labels from our method and the superiority of our method.

## References

[1] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012. 1

[2] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 12697–12705, 2019. 1

[3] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 1

[4] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *Porc. of IEEE Intl. Conf. on Computer Vision*, pages 9277–9286, 2019. 2

[5] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 652–660, 2017. 1

[6] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 770–779, 2019. 1

[7] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proc.*

*of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 567–576, 2015. 1

[8] He Wang, Yezhen Cong, Or Litany, Yue Gao, and Leonidas J Guibas. 3dioumatch: Leveraging iou prediction for semi-supervised 3d object detection. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 14615–14624, 2021. 1, 2

[9] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 1