Supplementary for "Exploring Geometry-aware Contrast and Clustering Harmonization for Self-supervised 3D Object Detection"

1. Dataset and Evaluation

To evaluate the performance of GCC-3D, we do experiments on nuScenes, Waymo and KITTI datasets. Unless otherwise noted, settings are the same for all experiments.

1.1. Detection on nuScenes

We first conduct experiments on nuScenes [2] dataset, which has 28k, 6k, 6k, annotated frames for training, validation, and testing, respectively. nuScenes uses a 32 lanes Lidar, which produces approximately 30k points per frame. To enrich input information and enable a more reasonable velocity estimation, a common practice[2] in nuScenes is transforming and accumulating LiDAR sweeps of non-annotated frames into its following annotated frame as input. The dataset has a severe imbalance among 10 classes, so that class-balanced grouping and sampling is adopted during supervised fine-tuning, following [18]. For experiments on nuScenes, we use a detection range of [-51.2m, 51.2m] for the X and Y axis, and [-5m, 3m] for Z axis. CenterPoint-voxel uses a (0.1m, 0.1m, 0.2m) voxel size and CenterPoint-pillar uses a (0.2m, 0.2m) grid. Following the training setting in [15], we optimize the model using AdamW[7] optimizer with one-cycle learning rate policy, with max learning rate $1e-3$ for VoxelNet and 0.002 for Pointpillar, weight decay 0.01, and momentum [0.85, 0.95].

Evaluation Metric For 3D detection evaluation, we adopt the official metrics of nuScenes dataset, that is, mean Average Precision (mAP) and nuScenes detection score [2] (NDS). The mAP uses a bird's-eye-view center distance $<$ 0.5m, 1m, 2m, 4m instead of standard box-overlap. NDS is a weighted average of mAP, attributes metrics including translation, scale, orientation, velocity, and other box attributes [2]. For 3D tracking, AMOTA[12] is used as criteria:

$$MOTA_r = 1 - \frac{IDS_r + FP_r + FN_r - (1-r)P}{rP},$$

$$AMOTA = \frac{1}{n} \sum_{r \in \{\frac{1}{n}, \frac{2}{n}, \ldots, 1\}} max(0, MOTA_r),$$

where r is a recall threshold, IDS_r, FP_r, FN_r are the number of ID switches, false positives, and false negatives until the top-scored detections reaching recall r, respectively. P is the total number of annotated objects in the dataset, and $n = 40$.

1.2. Detection on Waymo

Waymo Open Dataset [10] contains 798 training sequences and 202 validation sequences for vehicles and pedestrians. The point-clouds are captured with a 64 lanes Lidar, which produces about 180k Lidar points every 0.1s. Our Waymo model uses a detection range of [−75.2m, 75.2m] for the X and Y axis, and [−2m, 4m] for the Z axis. CenterPoint-Voxel uses a (0.1m, 0.1m, 0.15m) voxel size following PV-RCNN [8] while CenterPoint-Pillar uses a grid size of (0.32m, 0.32m). We use the same training schedule with nuScenes with an initial learning rate 3e-3.

Evaluation Metric We adopt the official released evaluation tools for evaluating our method, where the mean average precision (mAP) and the mean average precision weighted by heading (mAPH) are used for evaluation. The rotated IoU threshold is set as 0.7. We report LEVEL 2 mAP/mAPH for all experiments that denotes the ground-truth objects with at least 1 inside point.

1.3. Detection on KITTI

Transfer experiments have been done on the limited KITTI [4] dataset to evaluate generalization. KITTI contains 7481 training samples and 7518 testing samples. We follow the frequently used train/val split mentioned in [8] to divide the training samples into train split (3712 samples) and val split (3769 samples). We train the model with the batch size 24, learning rate 0.01 for 80 epochs on 8 V100 GPUs.

Evaluation Metric All results are evaluated by the mean average precision with a rotated IoU threshold 0.7 for cars and 0.5 for pedestrians and cyclists. The validation results are calculated with 11 recall positions to compare with the results by the previous works.

initial.	0.05		0.1		0.5		1	
	AMOTA↑	AMOTP↓	AMOTA↑	AMOTP↓	AMOTA↑	AMOTP↓	AMOTA↑	AMOTP↓
random init.	34.57	83.44	50.01	69.29	61.70	61.27	63.66	60.59
GCC-3D	35.47	83.14	51.16	68.97	62.31	61.08	64.17	60.53

Table 1. 3D tracking with limited labels on nuScenes val set evaluated with AMOTA and AMOTP metric. "random init." denotes random initilization baselines. ↑ is for higher better and ↓ is for lower better

2. Implement Details

2.1. GCC-3D Pre-train Setup

Architecture Our implementation is based on the open-sourced code of CenterPoint[15], we experiment with both VoxelNet[17] encoder and PointPillars [6] encoder.

In the Geometry-aware Contrast pre-train module, the initial voxel/pillar-wise feature will be passed through a two-layer MLP (with dimensions 128, 64) to project into latent space, with batch norm and ReLU. The latent space feature will be concatenated with the initial feature and passed through a one-layer MLP with dimension 64. We use the obtained feature for contrast pre-train. During fine-tune stage, the initial voxel/pillar-wise feature will be used. In Harmonized Clustering module, RoIAligned features of each pseudo instance will be passed through a two-layer MLP (with dimension 192, 128), with a batch norm, ReLU to project the feature to latent space for cluster pre-train. This two-layer MLP will not be used in fine-tune stage.

Multi-view Augmentation Setup We generate multi-views of the origin scene by random flip, scaling with a scale factor sampled from [0.95, 1.05] and rotation around vertical yaw axis between [-10, 10] degrees. We also do downsampling by a factor sampled from [0.9, 1].

Pseudo-instance Generation With available motion voxel set, we use morphology [1, 9] operation to get motion instance patches. First, we convert indexes in motion voxel set to a binary image (the image size is x-z point cloud range/voxel size). All motion voxel localizations have 1 and others have 0. Second, two opening operators with 3×1 and 1×3 kernels are used on binary images to smooth area contours and break narrow discontinuities. Finally, we choose the connected area of the top $K = 50$ point density as motion instance patches.

2.2. Data-Efficient 3D Object Detection Benchmark

During the fine-tuning stage, we conduct data augmentation of random flip, scaling with a scale factor sampled from [0.95, 1.05] and rotation around vertical yaw axis between [-10, 10] degrees. We also use ground-truth sampling, which copies and pastes points inside an annotated box from one frame to another frame. For nuScenes dataset, we fine-tune the models for 20 epochs. For Waymo dataset, we fine-tune the models for 12 epochs on VoxelNet and 36 epochs on Pointpillar. All models are trained with batch size 6 on 8 V100 GPUs.

Deep Cluster	Harmonization Term	Motion Patch	nuScenes mAP	nuScenes delta
random init.			25.79	-
✓			27.84	+2.05
✓	✓		28.29	+2.50
✓		✓	28.93	+3.14
✓	✓	✓	30.32	+5.77

Table 2. Comparison with different patch selection strategies. All results are finetuned on 5% nuScenes with Centerpoint-pp.

2.3. Compare with SOTA Experiments Setup

The results of SECOND, PART^2 and PV-RCNN in Table 3 (main body) are based on codebase OpenPCDet [11]. For fair comparison on Waymo, we use the same training schedule for 20% labeled dataset and train the model for 30 epochs following settings in OpenPCDet.

2.4. Transfer Experiments Setup

To evaluate the transfer capacity of our GCC-3D pre-training, we fine-tune on different datasets (KITTI, nuScenes and Waymo) and different models, including PV-RCNN [8], SECOND [13] and Centerpoint [15].

KITTI We fine-tune on the whole training set with the task-specific head of PV-RCNN [8] and SECOND [13] following the official codebase PCDet [11]. The training batch size is 24, learning rate is 0.01. The models are trained for 80 epochs on 8 V100 GPUs. Specifically, we initialize the 3D sparse convolution and RPN of PV-RCNN by pre-trained encoder φ and ϕ from GCC-3D and the Voxel Set Abstraction Module is randomly initialized. For SECOND, weights can be loaded from GCC-3D except for final classification and regression layers. Note that pretraining from nuScenes has unmatched point feature dimension due to extra relative timestamp of accumulating multi-sweeps point cloud. Therefore, the point feature is made up 0 when we load nuScenes pre-trained weights.

nuScenes We use a single sweep point cloud to avoid the above-mentioned problem of inconsistent input dimension. The point cloud is so sparse that the result is poor.

Waymo As the same with KITTI setup, we make up input dimension to be consistent with nuScenes. We fine-tune on 100% labeled training set based on Centerpoint-voxel and follow the training settings in Data-Efficient 3D Object Detection Benchmark (Section 1.1).

Method	Car			Pedestrian			Cyclist		
	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
SECOND	88.46	**78.57**	**77.42**	61.15	55.32	50.57	81.98	68.19	63.45
GCC-3D from nusc (SECOND)	88.16	78.41	77.25	**61.50**	**56.56**	**51.9**	**86.01**	**70.07**	**65.37**
GCC-3D from wm (SECOND)	**88.54**	78.52	77.41	60.59	54.88	49.98	83.89	68.56	63.94
PV-RCNN	88.94	79.12	78.52	65.21	58.99	54.29	86.89	**71.20**	**66.23**
GCC-3D from nusc (PV-RCNN)	**89.13**	83.59	78.77	65.24	57.53	54.07	**87.11**	71.14	65.77
GCC-3D from wm (PV-RCNN)	89.08	**83.67**	**78.80**	**66.14**	**60.34**	**56.04**	86.81	69.78	65.36

Table 3. Comparison with random initialization on KITTI validation set. "from nusc" indicates pre-training on nuScenes dataset. "from wm" means pre-trained weights from Waymo dataset.

Model	All		Car	Truck	CV.	Bus	Trailer	Barrier	Motor.	Bicycle	Ped.	TC.
	mAP	NDS										
WYSIWYG [5]	35.0	41.9	79.1	30.4	7.1	46.6	40.1	34.7	18.2	0.1	65.0	28.8
3DSSD [14]	42.6	56.4	81.2	47.2	12.6	61.4	30.5	47.9	36.0	8.6	70.2	31.1
HotSpotNet [3]	50.6	59.8	83.3	52.7	15.3	63.7	35.3	52.0	53.7	25.5	74.8	50.3
CBGS [18]	50.6	62.3	-	-	-	-	-	-	-	-	-	-
centerpoint-pp [15]	49.6	60.2	83.9	50.1	12.0	61.4	31.3	60.1	44.2	19.0	78.7	55.4
centerpoint-voxel [15]	56.2	64.5	84.8	53.9	16.8	67.0	35.9	64.8	55.8	36.4	83.1	63.4
GCC-3D (centerpoint-pp)	**50.8**	**60.8**	84.4	52.8	12.3	62.4	32.3	61.0	47.4	21.9	79.2	54.8
GCC-3D (centerpoint-voxel)	**57.3**	**65.0**	**85.0**	**54.7**	**17.6**	**67.2**	35.7	**65.0**	**56.2**	36.0	82.9	**63.7**

Table 4. Comparison with state-of-the-art on nuScenes. We show the NDS, mAP, and AP for each class. Abbreviations: construction vehicle (CV.), pedestrian (Ped.), motorcycle (Motor.), and traffic cone (TC.).

2.5. Other SSL Methods Setup

The training and optimization setup of PointContrast pre-train follow the same setup as Geometry-aware Contrast pre-train. And the setup of SwAV and Deepcluster follow the same setup as Harmonized Clustering pre-train.

3. More Quantity Results

3.1. Ablation Study on Instance Clustering

In harmonized clustering stage, we need ego-motion information to provide pseudo instances. However, these instances are not necessarily generated by moving patches and can be generated by random sampling or traditional clustering methods. Therefore, our module can be easily adopted on datasets without ego-motion. We did experiments to evaluate the effectiveness of harmonized clustering without the pseudo-instance proposal. To do that, we replace motion patches by randomly cropping patches from the dense point area of the scene. These differently selected patches are used for Deepcluster pre-train and harmonized clustering pre-train. We test these models on nuScenes dataset with 5% of supervised data during fine-tune stage. The results in Table 2 can be observed that both the harmonization term and motion proposal module help boost the performance (mAP 28.29% *v.s* 30.32% and mAP 28.93% *v.s* 30.32%). In particular, even though we randomly select patches from the scenes without ego-motion, our harmonized clustering model can still boost the performance significantly (mAP 25.79% *v.s* 28.93%).

3.2. Data-Efficient Object Tracking

To further evaluate pre-training performance on object tracking, we predict the positional difference of each detected object between the current and the past frame, and produce an additional regression term v during the supervised fine-tuning phrase, following [15, 16]. Then, detected objects in the current frame can be associated with past ones using closest distance matching and are kept until unmatched tracking up to T = 3 frames. Table 1 shows our unsupervised pre-training outperforms random initialization in object tracking task.

3.3. 3D Object Detection on KITTI

We show the detailed transfer results on KITTI for car, pedestrian and cyclist in Table 3.

3.4. 3D Object Detection on nuScenes across Classes

We show the detailed comparison, APs of each class, among different methods in Table 4. Our GCC-3D draws better performance compared to all current state-of-the-art methods on nuScenes by a large margin. Not only on mAP, it also outperforms those methods on AP of each class.

4. Quality Results

We show the quality comparisons of the train from scratch (denoted "baseline") Centerpoint-pp [15] and our

GCC-3D pre-training (indicated "GCC-3D") on 10% labeled nuScenes in Fig.1 and 2. The GCC-3D is more accurate than the baseline due to the help of discriminative spatial and semantic features. For example, random initialization produces more false positives than the proposed GCC-3D pre-training in Fig.1. We also show the quality comparisons based on Centerpoint-voxel [17] in Fig.3.

References

[1] Wilhelm Burger and Mark J Burge. *Principles of digital image processing: core algorithms*. Springer Science & Business Media, 2010. 2.1

[2] 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*, pages 11621–11631, 2020. 1.1

[3] Qi Chen, Lin Sun, Zhixin Wang, Kui Jia, and Alan Yuille. object as hotspots. In *ECCV*, 2020. 2.4

[4] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 1.3

[5] Peiyun Hu, Jason Ziglar, David Held, and Deva Ramanan. What you see is what you get: Exploiting visibility for 3d object detection. In *CVPR*, pages 11001–11009, 2020. 2.4

[6] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, pages 12697–12705, 2019. 2.1, 1, 2

[7] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 1.1

[8] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1.2, 1.3, 2.4

[9] Pierre Soille. *Morphological image analysis: principles and applications*. Springer Science & Business Media, 2013. 2.1

[10] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020. 1.2

[11] OpenPCDet Development Team. Openpcdet: An open-source toolbox for 3d object detection from point clouds. https://github.com/open-mmlab/OpenPCDet, 2020. 2.3, 2.4

[12] Xinshuo Weng and Kris Kitani. A baseline for 3d multi-object tracking. *arXiv preprint arXiv:1907.03961*, 2019. 1.1

[13] Y. Yan, Y. Mao, and B. Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10), 2018. 2.4

[14] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2.4

[15] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. *arXiv:2006.11275*, 2020. 1.1, 2.1, 2.4, 2.4, 3.2, 4

[16] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. *ECCV*, 2020. 3.2

[17] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *CVPR*, pages 4490–4499, 2018. 2.1, 4, 3

[18] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced grouping and sampling for point cloud 3d object detection. *arXiv preprint arXiv:1908.09492*, 2019. 1.1, 2.4

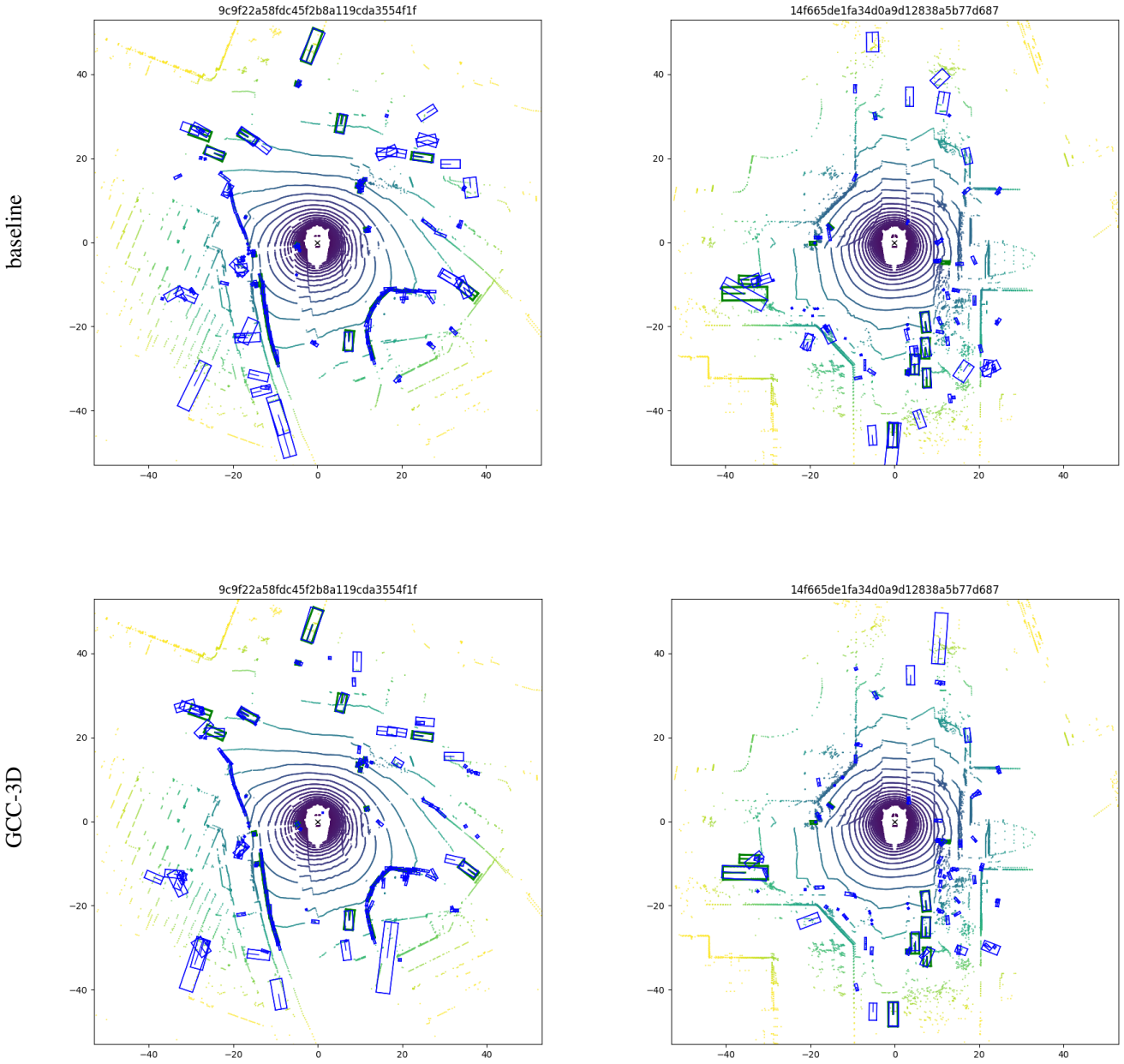

Figure 1. Qualitative results comparison on nuScenes val set with 10% labeled data. We compare the random initialization (baseline) and our GCC-3D pre-training (GCC-3D) based on PointPillars[6]. Our method is more accurate than the baseline method due to the help of the spatial sensitive and semantic representation. The green boxes are the groundtruth and the blue ones present the predicted results.

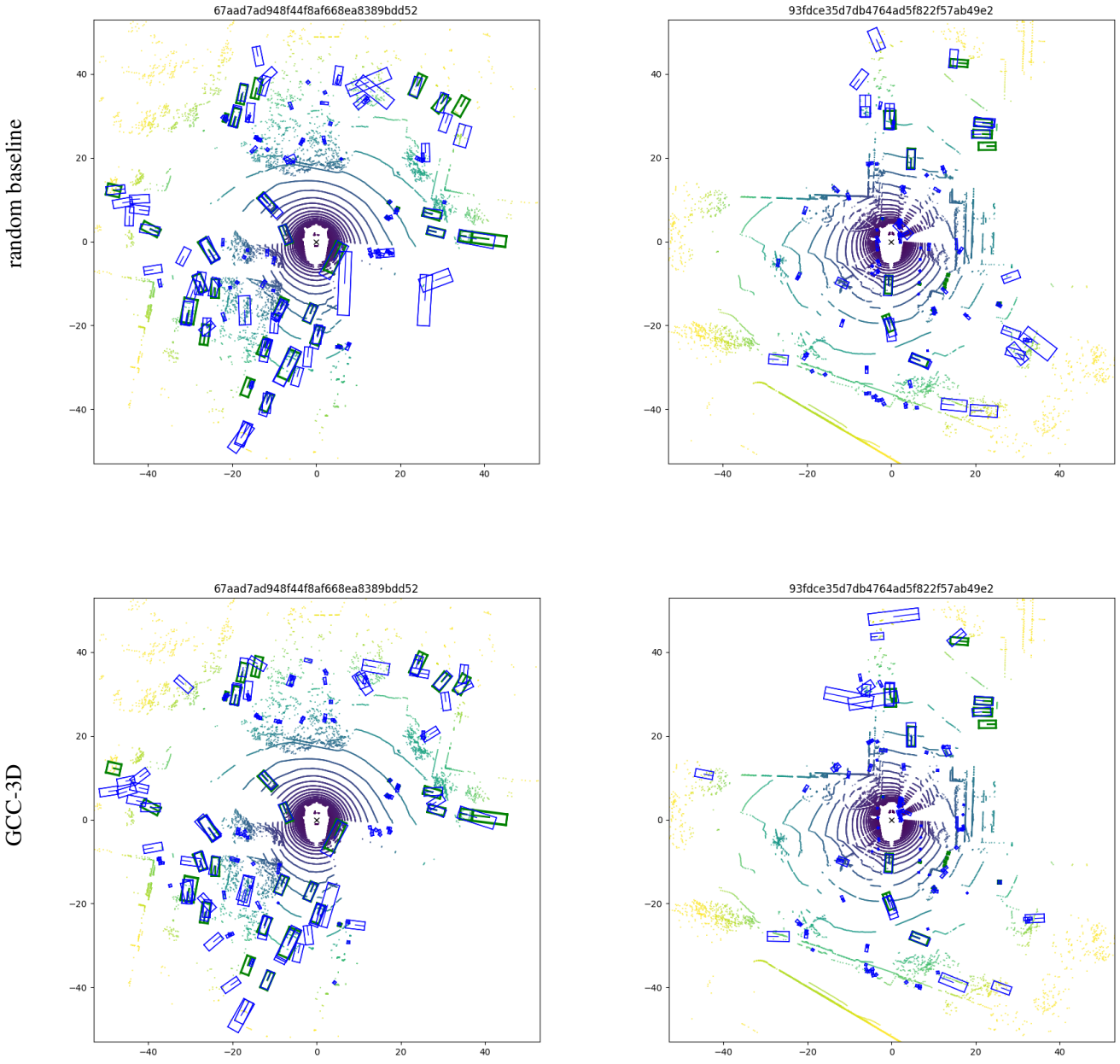

Figure 2. Qualitative results comparison on nuScenes val set with 10% labeled data. We compare the random initialization (baseline) and our GCC-3D pre-training (GCC-3D) based on PointPillars[6]. The green boxes are the groundtruth and the blue ones present the predicted results.

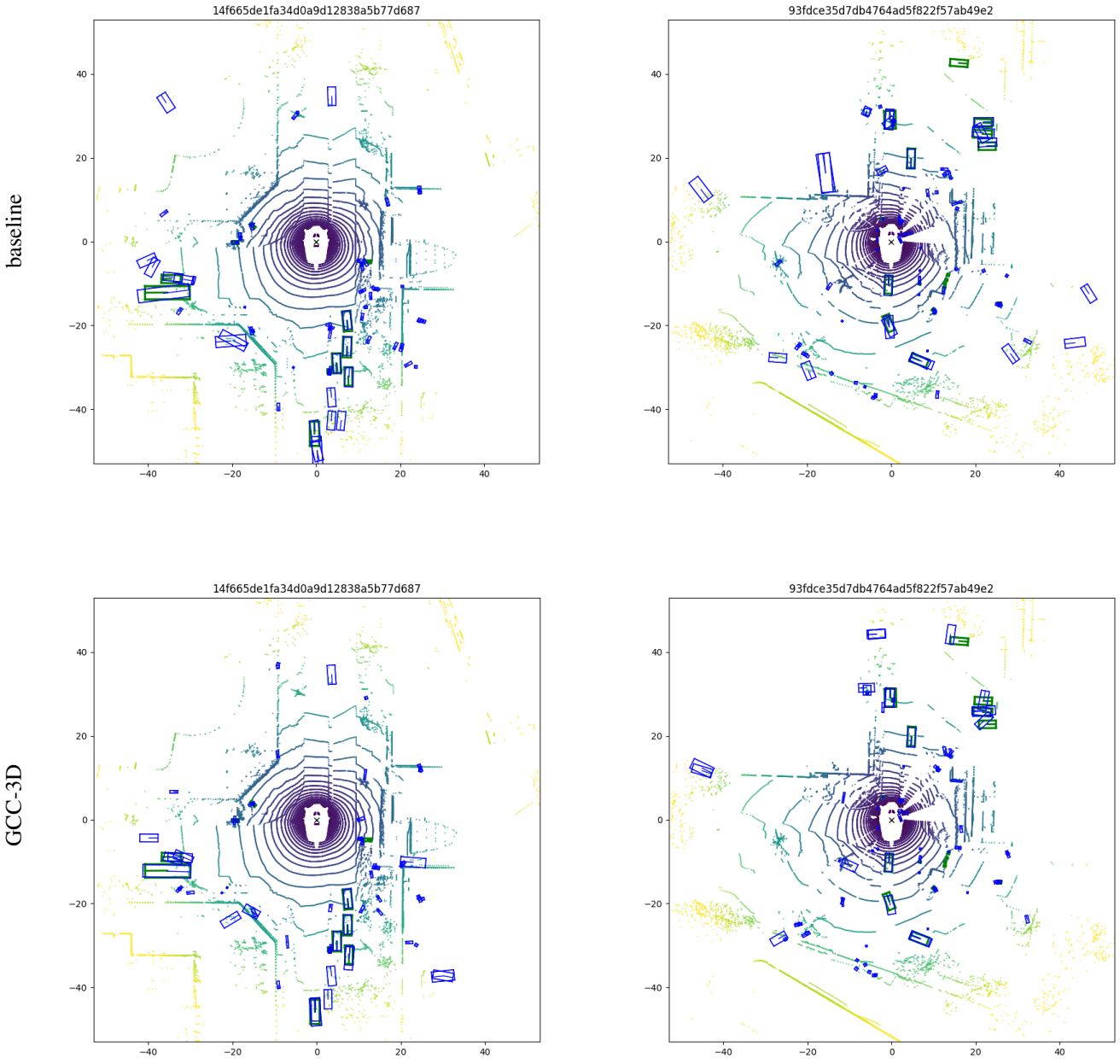

Figure 3. Qualitative results comparison on nuScenes val set with 10% labeled data. We compare the random initialization (baseline) and our GCC-3D pre-training (GCC-3D) based on VoxelNet[17]. The green boxes are the groundtruth and the blue ones present the predicted results.