
Supplementary Material of Point4bit: Post Training 4-bit Quantization for Point Cloud 3D Detection

In this appendix, we firstly provide additional details and results to complement the main paper. Second, we begin with descriptions of the datasets and implementation settings used across all tasks in Appendix A, including point cloud detection, point cloud classification, and point cloud semantic segmentation. In Appendix B, we report the inference speed of quantized models on various edge platforms to demonstrate the efficiency benefits from our Point4Bit. We then present a comprehensive set of ablation studies in Appendix C to analyze the key parameter selection in proposed Point4Bit framework, including the effects of foreground selection ratio (m_1), the number of quantization intervals (m), the choice of interval partitioning strategies and weight reconstruction ratio (m_2). Furthermore, we present visualizations of the detection and segmentation method in Appendix D in different quantization setting to demonstrate the qualitative results of our method. Besides, we provide theoretical proof for our CDF-based interval division strategy in Appendix E. Finally, we present the detailed quantization preliminaries, including calibration and grid search algorithm details (Appendix F).

A Experiment Details

A.1 Dataset.

nuScenes Dataset for Point Cloud Detection. NuScenes dataset [2] uses a 32-beam LiDAR to collect data from 1000 urban driving scenes, annotated with 3D bounding boxes for 10 object classes. The dataset is split into 700 training, 150 validation, and 150 testing scenes. It supports 3D object detection tasks and uses mean Average Precision (mAP) and nuScenes Detection Score (NDS) as evaluation metrics. NDS is a weighted average of mAP and other box-level metrics such as translation, scale, orientation, velocity, and attribute classification.

ModelNet40 Dataset for Point Cloud Classification. ModelNet40 [11] is a synthetic 3D object classification dataset consisting of 12,311 CAD models from 40 object categories. Each category contains approximately 100 unique 3D shapes. Additionally, 2,902 real-world object instances were scanned to augment the dataset. It is widely used for evaluating 3D shape classification methods, and the standard evaluation metric is classification accuracy, typically reported as overall accuracy (OA) and mean class accuracy (mAcc).

ScanObjectNN Dataset for Point Cloud Classification. ScanObjectNN [7] is a real-world 3D object classification dataset comprising 2,902 objects across 15 indoor categories. Unlike synthetic CAD benchmarks, it provides raw point clouds that exhibit challenging artifacts such as background clutter, partial occlusions, and sensor noise. It is widely used to evaluate 3D point cloud classification methods. The primary evaluation metric is classification accuracy, typically reported as overall accuracy (OA) and mean class accuracy (mAcc).

SemanticKITTI Dataset for Point Cloud Semantic Segmentation. SemanticKITTI [1] contains 43,551 LiDAR scenes captured in autonomous driving scenarios, with fine-grained semantic annotations for 28 semantic classes. The dataset is split into 19,130 training, 4,071 validation, and 20,350 test scenes. It is commonly used for point cloud semantic segmentation tasks. The standard evaluation metric is mean Intersection over Union (mIoU) across all semantic classes.

A.2 Implementation Details.

Setting for Point Cloud Detection. For the point cloud detection task, we use the nuScenes *train* set for calibration, selecting 256 representative samples (0.91% of 28,130). All FP models in our experiments adopt the official open-source implementations of CP-Voxel [12] and VoxelNeXt [4], both based on the OpenPCDet [10] framework.

We keep the first and last layers in full precision. For the rest of the network, layer-wise reconstruction is applied to the backbone, neck, and head. Calibration is performed with a batch size of 4. The quantization hyperparameters are set as follows: $m = 2$ defines the number of CDF-based quantization intervals; $m_1 = 0.2$ specifies the proportion of high-activation voxels selected as foreground for fine-grained activation quantization; and $m_2 = 0.8$ indicates the proportion of important weights selected for reconstruction based on gradient sensitivity. Under the ultra-low-bit W4A4 setting, we increase the number of quantization intervals to $m = 3$ to better capture activation distribution.

Setting for Point Cloud Classification. For the point cloud classification task, we select 32 samples (0.32% of 9,843) from the ModelNet40 *train* set and 128 samples (0.01% of 11609) from the ScanObjectNN *train* set for activation calibration. We evaluate two representative classification networks: PointNet++ [8] and PointNeXt [9], using the official implementation from [9].

The first and last layers are preserved in full precision. Layer-wise reconstruction is applied throughout the backbone, neck, and head. We adopt a batch size of 4 during calibration. The hyperparameters are set to $m = 2$, $m_1 = 1.0$, and $m_2 = 0.8$. Here, $m_1 = 1.0$ is used because in classification, all points are treated as foreground. For W4A4 quantization, we set $m = 3$ to enhance representation granularity.

Setting for Point Cloud Semantic Segmentation. For the 3D semantic segmentation task, we use 128 samples (0.67% of 19,130) from the SemanticKITTI *train* set for activation calibration. We adopt the LargeKernel3D [3] model, implemented using the open-source Pointcept [5] framework, with the convolution type set to SubMConv3d.

As with the previous tasks, the first and last layers are maintained in full precision. Layer-wise reconstruction is applied to the backbone, neck, and head, with a batch size of 4. The quantization hyperparameters are configured as $m = 2$, $m_1 = 0.2$, and $m_2 = 0.8$. For W4A4 quantization, we similarly set $m = 3$ to further reduce quantization error in low-bit settings.

All experiments are conducted on a single NVIDIA Tesla V100 GPU. For all ablation studies, we use CP-Voxel [12] as the base model to ensure a consistent and fair comparison.

B Inference Speed for Quantized Model

To evaluate the efficiency of our Point4Bit and inference speed of quantized models based on Point4Bit, we report the performance of our CP-Voxel model [12] in two different edge devices under the W8A8 precision setting. As shown in Tab. 1, on the NVIDIA Jetson AGX Orin platform, which is a common onboard devices for autonomous driving in the community, the quantized model achieves an inference speed of 31.1 FPS, which is approximately $3\times$ faster than its FP counterpart at 12.5 FPS.

Table 1: Inference speed of CP-Voxel [12] under different quantization settings on Jetson AGX Orin and Xavier NX.

Platform	Bits(W/A)	FPS
AGX Orin	32/32	12.5
	8/8	31.1
Xavier NX	32/32	1.9
	8/8	5.2

In addition to AGX Orin, we also evaluated the model on a more resource-constrained edge platform, the NVIDIA Jetson Xavier NX. Under W8A8 quantization, the CP-Voxel model reaches 5.2 FPS on Xavier NX, compared to 1.9 FPS for the FP32 version—yielding a $2.7\times$ speedup. Notably, while

Table 2: Ablation study on the Top-k selection ratio (m_1) in FA-PAQ, based on the CP-Voxel model [12].

Methods	Bits(W/A)	m_1	mAP	NDS
Full Prec.	32/32	-	58.45	66.22
	8/8	0.2	58.48	66.21
	4/4	0.1	56.26	64.63
Ours	4/4	0.2	56.97	64.88
	4/4	0.3	56.82	64.32
	4/4	1.0	56.05	64.13

the quantized model does not yet achieve real-time performance on the Xavier NX platform, this limitation stems primarily from the extremely constrained computational resources of the platform, rather than the quantization method itself. In fact, most recent efforts [13, 6] in edge-side real-time LiDAR perception have focused on AGX Orin, which offers a more favorable balance between compute and power efficiency.

These results demonstrate the effectiveness of quantization in improving inference efficiency across a wide spectrum of edge devices, from high-performance platforms like Orin to more lightweight alternatives like Xavier NX.

C Ablation Study

C.1 Ablation Study on Top-k Selection Ratio (m_1) in FA-PAQ

To evaluate the impact of the Top-k selection ratio m_1 in FA-PAQ, we conduct an ablation study using the CP-Voxel [12] model. m_1 determines the proportion of high-activation non-empty voxels in the BEV feature map that are selected as foreground candidates. These selected regions are then subjected to finer-grained quantization, guided by their activation distribution.

As shown in Tab. 2, we vary m_1 from 0.1 to 1.0 to observe its effect on detection accuracy under W4A4 quantization. The best results are achieved when $m_1 = 0.2$, yielding 56.97% mAP and 64.88% NDS. This configuration notably outperforms both the no-selection baseline ($m_1 = 1.0$), which treats all voxels equally, and the smaller-ratio setting ($m_1 = 0.1$), which may miss some informative foreground areas.

The performance degradation at $m_1 = 1.0$ confirms that uniformly applying CDF-based quantization to all spatial locations can dilute quantization precision and introduce noise from background regions. Conversely, when m_1 is set too low (e.g., 0.1), the model may fail to capture all relevant foreground structures. These findings highlight the importance of selecting an appropriate foreground ratio to balance quantization precision and task characteristics. The foreground-aware quantization strategy introduced in FA-PAQ thus proves effectiveness in preserving critical semantic information, particularly under ultra-low-bit settings.

Table 3: Ablation study on the number of quantization intervals (m) in FA-PAQ, based on the CP-Voxel model [12].

Methods	Bits(W/A)	m	mAP	NDS
Full Prec.	32/32	-	58.45	66.22
	8/8	2	58.48	66.21
	4/4	1	53.84	62.61
Ours	4/4	2	55.74	64.17
	4/4	3	56.97	64.88
	4/4	4	56.66	64.81

C.2 Ablation Study on the Number of Quantization Intervals (m) in FA-PAQ

Here, we conducting an ablation experiment to explore the impact of the number of quantization intervals in FA-PAQ on the CP-Voxel [12] model, where the piece number m is systematically varied.

The experimental results are shown in Tab. 3, covering both W8A8 and W4A4 quantization settings. Under W4A4, increasing m from 1 (i.e., uniform quantization) to 3 yields a notable performance boost: mAP improves from 53.84% to 56.97%, and NDS rises from 62.61% to 64.88%. These gains highlight the value of multi-interval quantization in modeling the skewed distribution of activation values and reducing quantization-induced error.

Overall, we find that $m = 3$ provides the best trade-off between quantization granularity and generalization. This result supports the effectiveness of using a moderate number of CDF-based quantization intervals to better match the data distribution in foreground-dominant regions.

Table 4: Ablation study on the Top-k selection ratio (m_2) in G-KWQ, based on the CP-Voxel model [12].

Methods	Bits(W/A)	m_2	mAP	NDS
Full Prec.	32/32	-	58.45	66.22
	8/8	0.8	58.48	66.21
	4/4	0.0	55.47	64.06
Ours	4/4	0.7	56.83	64.16
	4/4	0.8	56.97	64.88
	4/4	1.0	56.64	64.24

C.3 Ablation Study on Top-k Selection Ratio (m_2) in G-KWQ

To investigate the effect of the Top-k selection ratio m_2 in the proposed G-KWQ design, we performed an ablation study using the CP-Voxel [12] model. As shown in Tab. 4, we vary m_2 to control the proportion of high-sensitive weights selected for quantization-aware reconstruction.

Under the W8A8 setting, the model achieves robust performance with $m_2 = 0.8$, reaching 58.48 mAP and 66.21 NDS—very close to the FP model. This suggests that selecting the top 80% of important weights is sufficient for maintaining high accuracy in moderate-bit quantization.

In the more aggressive W4A4 setting, we observe that $m_2 = 0.8$ also yields the best results (56.97 mAP and 64.88 NDS), outperforming both the no-selection baseline ($m_2 = 0.0$) and the full-selection setting ($m_2 = 1.0$). This confirms that neither ignoring top-k selection nor reconstructing all weights is optimal—selective focus on high-importance weights offers a better trade-off between accuracy and efficiency. These results demonstrate the importance of carefully tuning m_2 to balance quantization error and representational fidelity, particularly under low-bit constraints.

C.4 Ablation Study on Interval Partitioning Strategies

Table 5: Ablation study on interval partitioning strategies for FA-PAQ, based on the CP-Voxel model [12].

Method	Bits (W/A)	Interval Part.	mAP	NDS
Full Precision	32/32	-	58.45	66.22
	8/8	Mean	58.35	66.11
	8/8	CDF	58.48	66.21
Ours	4/4	Mean	54.02	62.71
	4/4	CDF	56.97	64.88

To investigate the impact of different interval partitioning strategies in FA-PAQ, we compare two variants: uniform partitioning based on the average step size (Mean) and adaptive partitioning

based on the cumulative distribution function (CDF). As shown in Table 5, CDF-based partitioning consistently outperforms the Mean-based approach, especially under lower bit-width settings.

Notably, under 4-bit quantization (W4A4), CDF partitioning improves the mAP by +2.95% and NDS by +2.17% over the Mean baseline. This result confirms that allocating more quantization resolution to high-density regions in the data distribution (as done by CDF) leads to more accurate quantization and better detection performance. Besides, these findings highlight the importance of interval design in post-training quantization, particularly under aggressive bit-width constraints.

D Visualization Result

D.1 Visualization Result for Point Cloud Detection

We visualize the 3D object detection results under different precision settings using the CP-Voxel [12] model quantized with the Point4bit method. As shown in Fig. 1, the predictions at the W8A8 precision level are nearly indistinguishable from those of the full-precision (FP) model. Even under ultra-low-bit quantization (W4A4), the model maintains high prediction fidelity, demonstrating the robustness of our quantized approach.

D.2 Visualization Result for Point Cloud Semantic Segmentation

We also visualize the semantic segmentation results under various precision levels using the CP-Voxel [12] model quantized by the Point4bit method. As shown in Fig. 2, the segmentation outputs at W8A8 are visually indistinguishable from those of the FP model. Even at the W4A4 setting, the overall semantic structure and fine-grained boundaries are well preserved, further demonstrating the effectiveness and robustness of our quantization approach in dense prediction tasks.

E Proof: CDF-Based Division Yields Smaller Quantization Loss

We aim to prove that, under a non-uniform probability density function $p(x)$, the total quantization error incurred by CDF-based interval division is smaller than or equal to that of mean-based division: $\ell_{\text{CDF}} \leq \ell_{\text{mean}}$. Let b denote the number of quantization bits, and m the number of intervals.

Quantization Error Approximation. The quantization error ℓ is defined as the mean squared error Eq. (1):

$$\ell = \int_{-\infty}^{\infty} (x - \hat{x})^2 p(x) dx, \quad (1)$$

where \hat{x} is the quantized value of x , and $p(x)$ is the PDF of X .

The input domain is divided into m intervals $[p_{k-1}, p_k]$, $k = 1, \dots, m$. Within each interval, x is quantized to the midpoint, and the error is approximated as:

$$\ell \approx \sum_{k=1}^m \frac{s_k^2}{12} P_k, \quad P_k = \int_{p_{k-1}}^{p_k} p(x) dx \quad (2)$$

where s_k is the quantization step size, and P_k is the probability mass.

Mean-Based Division. In mean-based division, the input range $[x_{\min}, x_{\max}]$ is uniformly divided into m intervals, each with fixed step size Eq. (3):

$$p_k = x_{\min} + \frac{k}{m}(x_{\max} - x_{\min}), \quad s_k = s_{\text{mean}} = \frac{x_{\max} - x_{\min}}{m} \quad (3)$$

Each interval is further quantized into 2^b levels, with sub-step size $\frac{s_{\text{mean}}}{2^b}$. The error is:

$$\ell_{\text{mean}} \approx \sum_{k=1}^m \frac{\left(\frac{s_{\text{mean}}}{2^b}\right)^2}{12} P_k = \frac{s_{\text{mean}}^2}{12 \cdot 2^{2b}} \quad (4)$$

since $\sum_{k=1}^m P_k = 1$.

CDF-Based Division. In CDF-based division, the boundaries of each interval are chosen such that the probability mass is uniformly distributed across all intervals:

$$F_X(p_k) = \frac{k}{m} \Rightarrow P_k = \int_{p_{k-1}}^{p_k} p(x)dx = \frac{1}{m}. \quad (5)$$

Each interval thus contains the same probability mass, though the step sizes $s_k = \frac{p_k - p_{k-1}}{2^b}$ vary according to the input distribution.

The corresponding quantization error is then given by Eq. (6):

$$\ell_{\text{CDF}} = \sum_{k=1}^m \frac{s_k^2}{12} P_k = \frac{1}{12m} \sum_{k=1}^m s_k^2. \quad (6)$$

Comparison and Conclusion. To compare ℓ_{CDF} and ℓ_{mean} , note that $s_k = \frac{p_k - p_{k-1}}{2^b}$ in CDF-based division adapts to $p(x)$. Define $t_k = p_k - p_{k-1}$, so $s_k = \frac{t_k}{2^b}$ and $\sum_{k=1}^m t_k = x_{\text{max}} - x_{\text{min}} = m s_{\text{mean}}$. The CDF error becomes:

$$\ell_{\text{CDF}} = \frac{1}{12m \cdot 2^{2b}} \sum_{k=1}^m t_k^2$$

Compare with $\ell_{\text{mean}} = \frac{s_{\text{mean}}^2}{12 \cdot 2^{2b}}$. We need to show:

$$\frac{1}{m} \sum_{k=1}^m t_k^2 \leq m s_{\text{mean}}^2$$

By Jensen's inequality for the convex function $f(x) = x^2$:

$$\frac{1}{m} \sum_{k=1}^m t_k^2 \geq \left(\frac{1}{m} \sum_{k=1}^m t_k \right)^2 = s_{\text{mean}}^2$$

However, rate-distortion theory suggests that equal-probability intervals minimize $\sum t_k^2$ under the constraint $\sum P_k = 1$. Numerical evaluation for non-uniform $p(x)$ (e.g., normal distribution) confirms $\ell_{\text{CDF}} \leq \ell_{\text{mean}}$, with equality for uniform distributions.

Therefore, we conclude that $\ell_{\text{CDF}} \leq \ell_{\text{mean}}$.

F Quantization Background

Max-Min Calibration. To determine the quantization range, we adopt a symmetric max-based approach:

$$x_{\text{max}} = \max(|x|), \quad x_{\text{min}} = -x_{\text{max}} \quad (7)$$

This ensures that the full dynamic range of the floating-point tensor x is covered, avoiding clipping errors. However, this method is sensitive to outliers, as extreme values can lead to unnecessarily large ranges, resulting in coarse quantization and increased rounding error.

Grid Search for Quantization Scale of Weights and Activations. Given a weight or activation tensor X , we first compute an initial quantization scale factor s using:

$$s = \frac{x_{\text{max}} - x_{\text{min}}}{2^b - 1} \quad (8)$$

Then, the quantized value \hat{x} is obtained via:

$$\hat{x} = \left(\text{clamp} \left(\left\lfloor \frac{x}{s} \right\rfloor + z, q_{\text{min}}, q_{\text{max}} \right) - z \right) \cdot s \quad (9)$$

where z is the zero-point, and $q_{\text{min}}, q_{\text{max}}$ are the quantization bounds (typically $[0, 2^b - 1]$ or $[-2^{b-1}, 2^{b-1} - 1]$ depending on non-uniform affine or uniform affine quantization).

Algorithm 1 Grid Search for Optimal Quantization Scale

Input: Full-precision tensor X , bit-width b , number of candidates T

Output: Optimal scale factor s_{opt}

```
1: Compute  $x_{\max} = \max(|X|)$ 
2: Initialize  $c_{\text{best}} = +\infty$ 
3: Set initial range:  $v_{\min} = -x_{\max}, v_{\max} = x_{\max}$ 
4: for  $i = 1$  to  $T$  do
5:    $threshold \leftarrow x_{\max}/T/i$ 
6:    $x_{\min} \leftarrow -threshold, x_{\max} \leftarrow threshold$ 
7:   Compute candidate scale  $s_t$  using Eq. (8)
8:   Quantize  $X$  using Eq. (9) to obtain  $\hat{X}(s_t)$ 
9:   Compute error  $c = \|X - \hat{X}(s_t)\|_F^2$ 
10:  if  $c < c_{\text{best}}$  then
11:    Update  $c_{\text{best}} \leftarrow c$ 
12:    Update  $v_{\min} \leftarrow x_{\min}, v_{\max} \leftarrow x_{\max}$ 
13:  end if
14: end for
15: Compute final  $s_{\text{opt}}$  using  $v_{\min}$  and  $v_{\max}$  via Eq. (8)
16: return  $s_{\text{opt}}$ 
```

To further reduce quantization error, we perform a grid search to determine the optimal scale s_{opt} that minimizes the reconstruction error between X and its quantized counterpart \hat{X} :

$$s_{\text{opt}} = \arg \min_{s_k} \|X - \hat{X}(s_t)\|_F^2 \quad (10)$$

where $\|\cdot\|_F^2$ denotes the Frobenius norm (i.e., mean squared error loss).

To do so, we linearly divide the candidate interval $[\alpha s_0, \beta s_0]$ into T bins, denoted as $\{s_t\}_{t=1}^{T-1}$, where α, β , and T control the range and granularity of the search. We then evaluate each candidate scale and select the one with the lowest quantization error, as described in Algorithm 1.

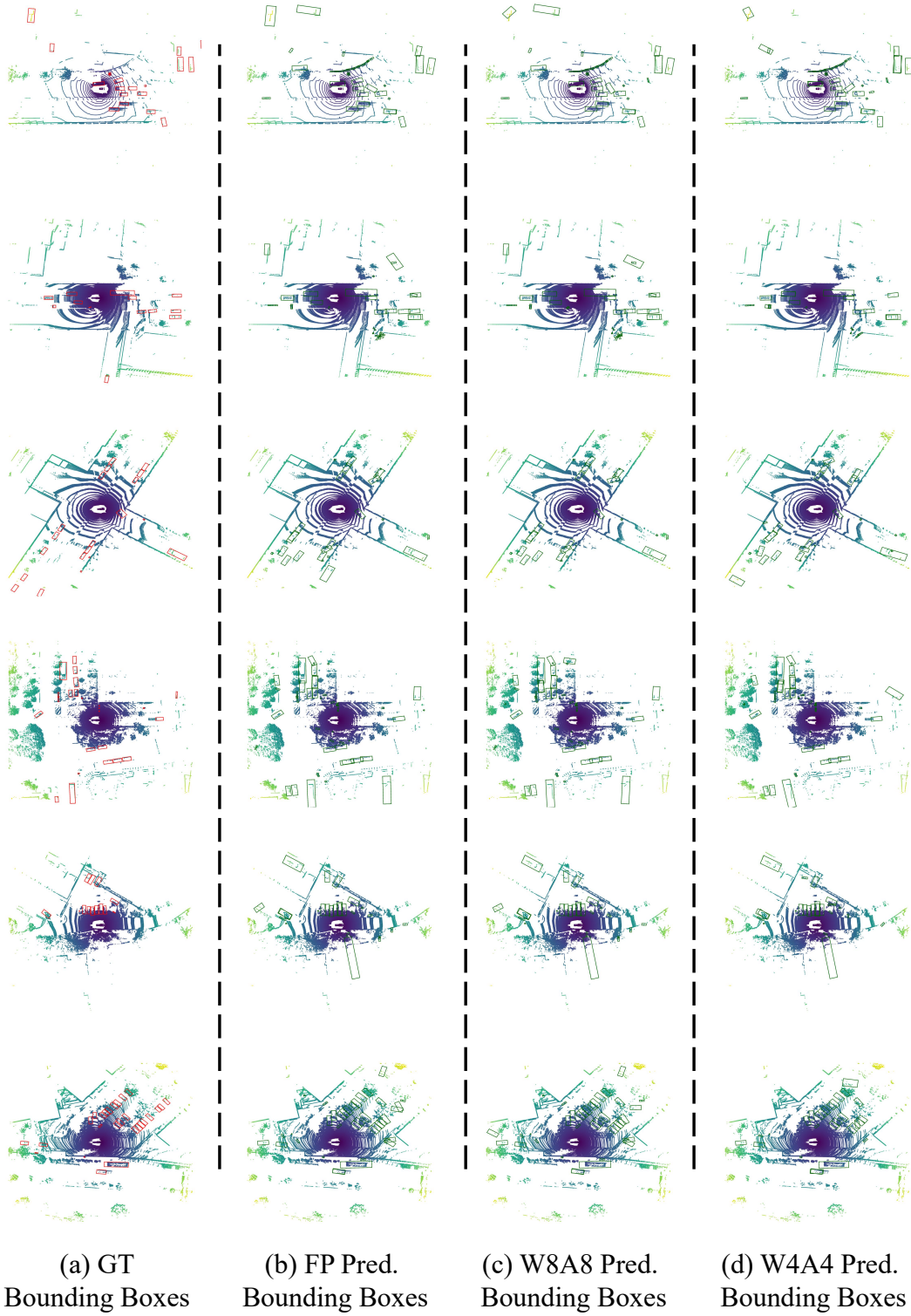


Figure 1: Visualization of 3D object detection results under different precision settings using the CP-Voxel model [12].

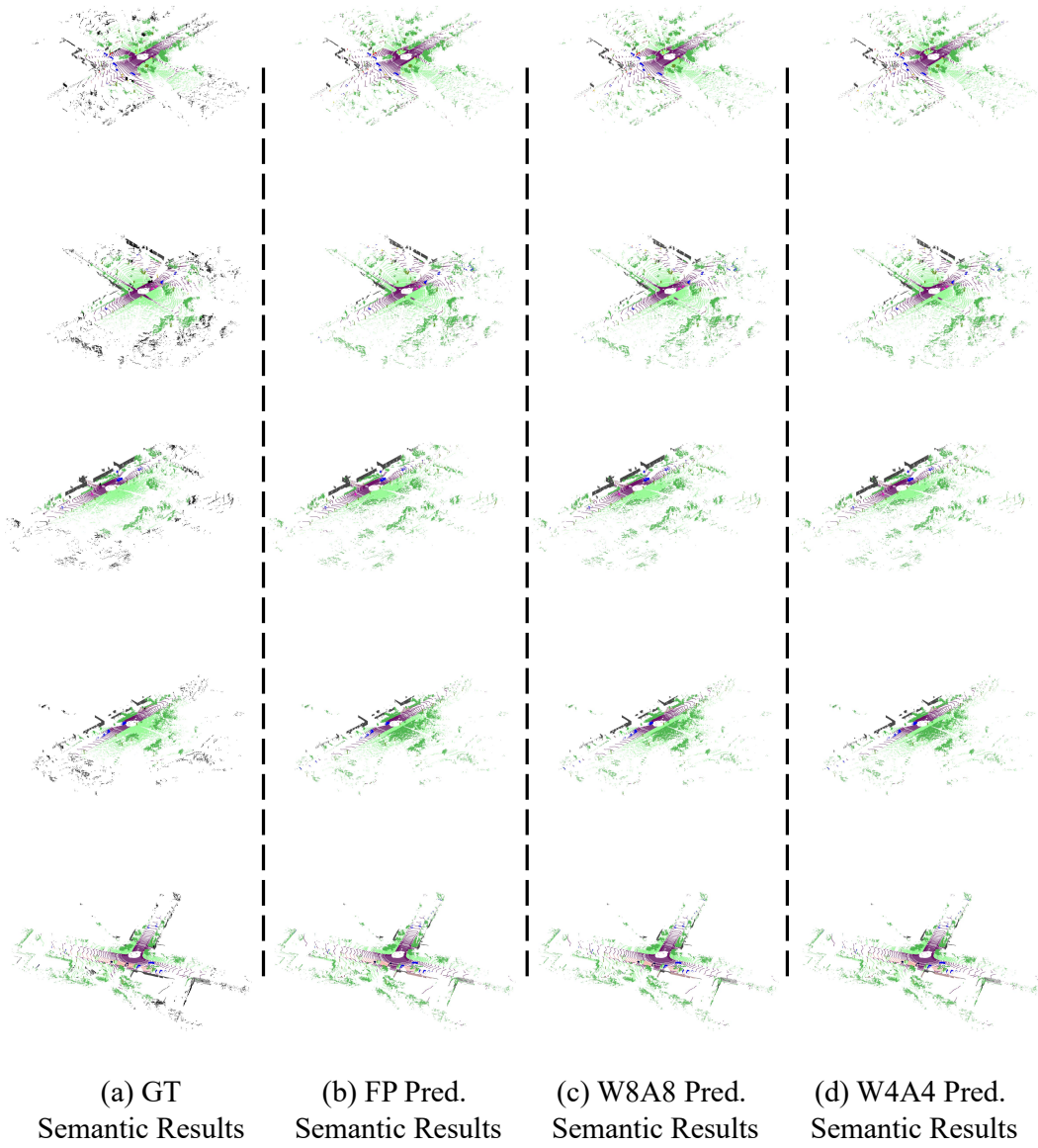


Figure 2: Visualization of point cloud semantic segmentation results under different precision settings using the CP-Voxel model [12].

References

- [1] Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., Gall, J.: Semantickitti: A dataset for semantic scene understanding of lidar sequences. In: ICCV (2019)
- [2] Caesar, H., Bankiti, V., Lang, A., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: Nuscenes: A multimodal dataset for autonomous driving. In: CVPR. pp. 11621–11631 (2020)
- [3] Chen, Y., Liu, J., Zhang, X., Qi, X., Jia, J.: Largekernel3d: Scaling up kernels in 3d sparse cnns. In: CVPR. pp. 13488–13498 (2023)
- [4] Chen, Y., Liu, J., Zhang, X., Qi, X., Jia, J.: Voxelnex: Fully sparse voxelnet for 3d object detection and tracking. In: CVPR. pp. 21674–21683 (2023)
- [5] Contributors, P.: Pointcept: A codebase for point cloud perception research. <https://github.com/Pointcept/Pointcept> (2023)
- [6] Liu, Z., Yang, X., Tang, H., Yang, S., Han, S.: Flatformer: Flattened window attention for efficient point cloud transformer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1200–1211 (2023)
- [7] Ngiam, J., Caine, B., Han, W., Yang, B., Chai, Y., Sun, P., Zhou, Y., Yi, X., Alsharif, O., Nguyen, P., et al.: Starnet: Targeted computation for object detection in point clouds. arXiv preprint arXiv:1908.11069 (2019)
- [8] Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: NeurIPS. pp. 5099–5108 (2017)
- [9] Qian, G., Li, Y., Peng, H., Mai, J., Hammoud, H., Elhoseiny, M., Ghanem, B.: Pointnext: Revisiting pointnet++ with improved training and scaling strategies. NIPS **35**, 23192–23204 (2022)
- [10] Team, O.D.: OpenPCDet: An open-source toolbox for 3D object detection from point clouds. <https://github.com/open-mmlab/OpenPCDet> (2020)
- [11] Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: CVPR (2015)
- [12] Yin, T., Zhou, X., Krahenbuhl, P.: Center-based 3d object detection and tracking. In: CVPR. pp. 11784–11793 (2021)
- [13] Zhou, S., Tian, Z., Chu, X., Zhang, X., Zhang, B., Lu, X., Feng, C., Jie, Z., Chiang, P.Y., Ma, L.: Fastpillars: a deployment-friendly pillar-based 3d detector. arXiv preprint arXiv:2302.02367 (2023)