LiDAR-PTQ: Post-Training Quantization for Point Cloud 3D Object Detection

Sifan Zhou[1,2*], Liang Li[2], Xinyu Zhang[2], Bo Zhang[2], Shipeng Bai[3*], Miao Sun[4]
Ziyu Zhao[1], Xiaobo Lu[1†], Xiangxiang Chu[2†‡]
[1]Southeast University [2]Meituan Inc [3]Zhejiang University [4]Nanyang Technological University
sifanjay@gmail.com, xblu2013@126.com, chuxiangxiang@meituan.com

Abstract

Due to highly constrained computing power and memory, deploying 3D lidar-based detectors on edge devices equipped in autonomous vehicles and robots poses a crucial challenge. Being a convenient and straightforward model compression approach, Post-Training Quantization (PTQ) has been widely adopted in 2D vision tasks. However, applying it directly to 3D lidar-based tasks inevitably leads to performance degradation. As a remedy, we propose an effective PTQ method called LiDAR-PTQ, which is particularly curated for 3D lidar detection (both SPConv-based and SPConv-free). Our LiDAR-PTQ features three main components, **(1)** a sparsity-based calibration method to determine the initialization of quantization parameters, **(2)** a Task-guided Global Positive Loss (TGPL) to reduce the disparity between the final predictions before and after quantization, **(3)** an adaptive rounding-to-nearest operation to minimize the layerwise reconstruction error. Extensive experiments demonstrate that our LiDAR-PTQ can achieve state-of-the-art quantization performance when applied to CenterPoint (both Pillar-based and Voxel-based). To our knowledge, for the very first time in lidar-based 3D detection tasks, the PTQ INT8 model's accuracy is almost the same as the FP32 model while enjoying $3\times$ inference speedup. Moreover, our LiDAR-PTQ is cost-effective being $30\times$ faster than the quantization-aware training method. Code will be released at `https://github.com/StiphyJay/LiDAR-PTQ`.

1 INTRODUCTION

LiDAR-based 3D detection has a wide range of applications in self-driving and robotics. It is important to detect the objects in the surrounding environment fastly and accurately, which places a high demand for both performance and latency. Currently, mainstream grid-based 3D detectors convert the irregular point cloud into arranged grids (voxels/pillars), and achieve top-ranking performance (Jiageng Mao, 2023) while facing a crucial challenge when deploying 3D lidar-based models on resource-limited edge devices. Therefore, it is important to improve the efficiency of grid-based 3D perception methods (e.g., reduce memory and computation cost).

Quantization is an efficient model compression approach for high-efficiency computation by reducing the number of bits for activation and weight representation. Compared to quantization-aware training (QAT) methods, which require access to all labeled training data and substantial computation resources, Post-training quantization (PTQ) is more suitable for fast and effective industrial applications. This is because PTQ only needs a small number of unlabeled samples as calibration set. Besides, PTQ does not necessitate retraining the network with all available labeled data, resulting in a shorter quantization process. Although several advanced PTQ methods (Nagel et al., 2020; Li et al., 2021; Wei et al., 2022; Yao et al., 2022) have been proposed for RGB-based detection tasks, applying it directly to 3D lidar-based tasks inevitably leads to performance degradation due to the differences between images and point clouds.

As shown in Fig 1, the inherent sparsity and irregular distribution of LiDAR point clouds present new challenges for the quantization of 3D Lidar-based detectors. **(1)** The sparsity of point cloud.

[*]Work done as an intern at Meituan. [†] Corresponding author. [‡]Project leader.

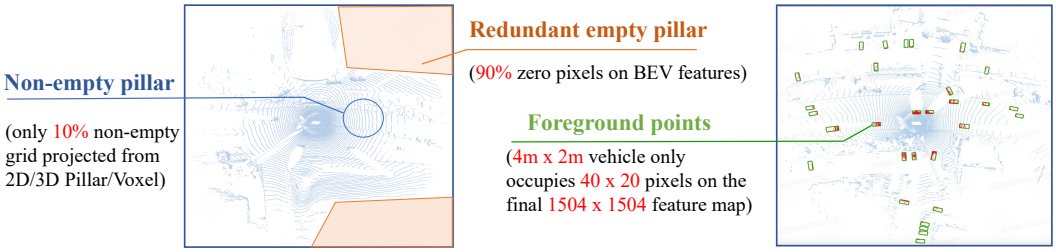


Figure 1: The sparsity of point cloud on 3D LiDAR-based object detection. Orange area means empty area, blue point means the point cloud (non-empty area) in a scenario, green box means the 3D Bboxes, and red point means foreground points.

Different from dense RGB images, non-zero pixels only occupy a very limited part of the whole scenario (about 10% in Waymo dataset (Sun et al., 2020)). For example, the huge number of zero pixel lead to significant differences in activation distribution compared to dense RGB-based tasks. **(2)** Larger arithmetic range. Compared with the 8-bit (0-255) RGB images, the point coordinates after voxelization are located in a $1504 \times 1504 \times 40$ (voxel size $= 0.1$m) 3D space in Waymo dataset, which makes it more susceptible to the effects of quantization (such as clipping error). **(3)** Imbalance between foreground instances and large redundant background area. For example, based on CenterPoint-Voxel (Yin et al., 2021), a vehicle with $4m \times 2m$ occupies only 40×20 pixels in the input 1504×1504 BEV feature map. Such small foreground instances and large perception ranges in 3D detection require the quantized model to have less information loss to maintain detection performance. Therefore, these challenges hinder the direct application of quantization methods developed for 2D vision tasks to 3D point cloud tasks.

To tackle the above challenge, we propose an effective PTQ method called LiDAR-PTQ, which is specifically curated for 3D LiDAR-based object detection tasks. Firstly, we introduce a sparsity-based calibration method to determine the initialization of quantization parameters on parameter space. Secondly, we propose Task-guided Global Positive Loss (TGPL) to find the quantization parameter on model space that is suitable for final output performance. Thirdly, we utilize an adaptive rounding value to mitigate the performance gap between the quantized and the full precision model. The proposed LiDAR-PTQ framework is a general and effective quantization method for both SPConv-based and SPConv-free 3D detection models. Extensive experiments on various datasets evaluate that our LiDAR-PTQ can achieve state-of-the-art quantization performance (Fig 2) when applied to CenterPoint (both Pillar-based and Voxel-based). To our knowledge, for the very first time in LiDAR-based 3D detection tasks, the PTQ INT8 model's accuracy is almost the same as the FP32 model while enjoying $3\times$ inference speedup. Moreover, our LiDAR-PTQ is cost-effective being $30\times$ faster than QAT method. We will release our code to the community.

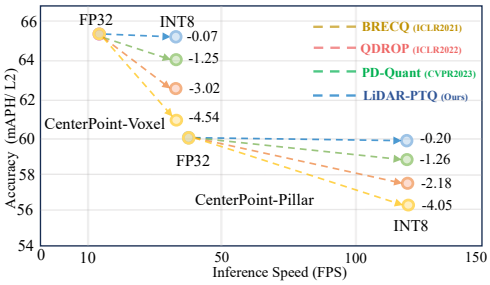


Figure 2: Performance comparison

Here, we summarize our main contributions as follows:

- Unveiling the root cause of performance collapse in the quantization of the 3D LiDAR-based detection model. Furthermore, we propose the sparsity-based calibration method to initialize the quantization parameter.
- TGPL: A Task-guided Global Positive Loss (TGPL) function to minimize the output disparity on model space which helps improve the quantized performance.
- LiDAR-PTQ: a general and effective quantization method for both SPConv-based and SPConv-free 3D detection models. Extensive experiments demonstrate LiDAR-PTQ can achieve state-of-the-art quantization performance on CenterPoint (both Pillar-based and Voxel-based).

- To our knowledge, for the very first time in LiDAR-based 3D detection tasks, the PTQ INT8 model's accuracy is almost the same as the FP32 model while enjoying $3\times$ inference speedup. Moreover, LiDAR-PTQ is cost-effective being $30\times$ faster than QAT method.

2 PRELIMINARIES

LiDAR-based 3D object detection. Given a point set with N points in the 3D space, which is defined as $\mathbf{P} = \{\mathbf{p}_i = [x_i, y_i, z_i, r_i]^T \in \mathbb{R}^{N\times 4}\}$, where x_i, y_i, z_i denote the coordinate values of each point along the axes X, Y, Z, respectively, and r_i is the laser reflection intensity. Given a set of object in the 3D scene $\mathbf{B} = \{\mathbf{b}_j = [x_j, y_j, z_j, h_j, w_j, l_j, \theta_j, c_j]^T \in \mathbb{R}^{M\times 8}\}$, where M is the total number of objects, b_i is the i-th object in the scene, x_j, y_j, z_j is the object's center, h_j, w_j, l_j is the object's size, θ_j is the object's heading angle and c_j is the object's class. The task of LiDAR-based 3D object detection is to detect the 3D boxes $\mathbf{B}$ from the point cloud $\mathbf{P}$ accurately.

Quantization for tensor. The quantization operation is defined as the mapping of a floating-point (FP) value x (weights or activations) to an integer value x_{int} according to the following equation:

$$x_{int} = clamp(\lfloor \frac{x}{s} \rceil + z, q_{min}, q_{max}) \tag{1}$$

where $\lfloor \cdot \rceil$ is the rounding-to-nearest operator, which results in the rounding error Δr. The function $clamp(\cdot)$ clips the values that lie outside of the integer range $[q_{min}, q_{max}]$, incurring a clipping error Δc. x_{int} represents the quantized integer value. z is zero-point. s denotes the quantization scale factor, which reflects the proportional relationship between FP values and integers. $[q_{min}, q_{max}]$ is the quantization range determined by the bit-width b. Here, we adopt uniform signed symmetric quantization, as it is the most widely used in TensorRT (Migacz, 2017) and brings significant acceleration effect. Therefore, $q_{min} = -2^{b-1}$ and $q_{max} = 2^{b-1} - 1$. Nonuniform quantization (Jeon et al., 2022) is challenging to deploy on hardware, so we disregard it in this work. Generally, weights can be quantized without any need for calibration data. Therefore, the quantization of weights is commonly solved using grid search or analytical approximations with closed-form solution (Banner et al., 2019; Nagel et al., 2021) to minimize the mean squared error (MSE) in PTQ. However, activation quantization is input-dependent, so often requires a few batches of calibration data for the estimation of the dynamic ranges to converge. To approximate the real-valued input x, we perform the de-quantization step:

$$\hat{x} = (x_{int} - z) \cdot s \tag{2}$$

where $\hat{x}$ is the de-quantized FP value with an error that is introduced during the quantization process.

Quantization range. If we want to reduce clipping error Δc, we can increase the quantization scale factor s to expand the quantization range. However, increasing s leads to increased rounding error Δr because Δr lies in the range $\left[-\frac{s}{2}, \frac{s}{2}\right]$. Therefore, the key problem is how to choose the quantization range (x_{min}, x_{max}) to achieve the right trade-off between clipping and rounding error. Specifically, when we set fixed bit-width b, the quantization scale factor s is determined by the quantization range:

$$s = (x_{max} - x_{min}) \,/\, \left(2^b - 1\right) \tag{3}$$

There are two common methods for quantization range setting.

i): *Max-min calibration.* We can define the quantization range as:

$$x_{max} = max(|x|), x_{min} = -x_{max} \tag{4}$$

to cover the whole dynamic range of the floating-point value x. This leads to no clipping error. However, this approach is sensitive to outliers as strong outliers may cause excessive rounding errors.

ii): *Entropy calibration.* TensorRT (Migacz, 2017) minimize the information loss between x and $\hat{x}$ based on the KL divergence to determine the quantization range:

$$\underset{x_{min}, x_{max}}{\arg\min} \; D_{KL}(x, \hat{x}) \tag{5}$$

where D_{KL} denotes the Kullback-Leibler divergence function. The entropy calibration will saturate the activations above a certain threshold to remove outliers. More details refer to the appendix.

Quantization for network. For a float model with N layer, we primarily focus on the quantization of convolutional layers or linear layers, which mainly involves the handling of weights and activations.

For a given layer L_i, we initially execute quantization operations on its weight and input tensor, as illustrated in Eq 1 and 2, yielding $\hat{W}_i$ and $\hat{I}_i$. Consequently, the quantized output of this layer can be expressed as follows.

$$\hat{A}_i = f(BN(\hat{I}_i \circledast \hat{W}_i) \tag{6}$$

where $\circledast$ denotes the convolution operator, $BN(\cdot)$ is the Batch-Normalization procedure, and $f(\cdot)$ is the activation function. Quantization works generally take into account the convolution, Batch Normalization (BN), and activation layers.

3 METHODOLOGY

Here, we first conduct PTQ ablation study on the CenterPoint-Pillar (Yin et al., 2021) model using two different calibrators (Entropy and Max-min) on Waymo val set. As shown in Table 1, when using INT8 quantization, the performance drop is severely compromised for both the calibration method, especially for the entropy calibrator with a significant accuracy drop of **-38.67 mAPH/L2**. However, directly employing the Max-min calibrator yielded better results, yet not unsatisfactory. It is entirely contrary to our experience in 2D model quantization, where entropy calibration effectively mitigates the impact of outliers, thereby achieving superior results (Nagel et al., 2021). Similar observations are also discussed in Stäcker et al. (2021). This anomaly prompts us to propose a general and effective PTQ method for 3D LiDAR-based detectors. In Waymo dataset, the official evaluation tools evaluated the methods in two difficulty levels: LEVEL_1 for boxes with more than five LiDAR points, and LEVEL_2 for boxes with at least one LiDAR point. Here we report the metrics in Mean Average Precision with Heading / LEVEL_2 (mAPH/L2), which is a widely adopted metric by the community.

Table 1: Ablation study.

Method	Bits(W/A)	LEVEL_2 mAPH		
		Mean	Vehicle	Pedestrian
Full Prec.	32/32	60.32	65.42	55.23
Entropy	8/8	21.65 **(-38.67)**	29.41 **(-36.02)**	11.89 **(-43.34)**
Max-Min	8/8	52.91 **(-7.41)**	55.37 **(-10.05)**	50.45 **(-4.78)**

3.1 LiDAR-PTQ Framework

In this paper, we propose a post-training quantization framework for point cloud models, termed LiDAR-PTQ. Our LiDAR-PTQ could enable the quantized model to achieve almost the same performance as the FP mode, and there is no extra huge computation cost and access to labeled training data. This framework primarily comprises three components.

i) Sparsity-based calibration: We employ a Max-min calibrator equipped with a lightweight grid search to appropriately initialize the quantization parameters for both weights and activations.

ii) Task-guided Global Positive Loss (TGPL): This component utilizes a specially designed foreground-aware global supervision to further optimize the quantization parameters of activation.

iii) Adaptive rounding-to-nearest: This module aims to mitigate the weight rounding error Δr by minimizing the layer-wise reconstruction error.

In summary, our method first initializes the quantization parameters for weights and activations through a search in the parameter space, and then further refine them through a process of supervised optimization in the model space. Consequently, our method is capable of achieving a quantized accuracy that almost matches their float counterparts for certain lidar detectors.

We formulate the our LiDAR-PTQ algorithm for a full precision 3D detector in Algorithm 1. Next, we will provide detailed explanations for these three parts.

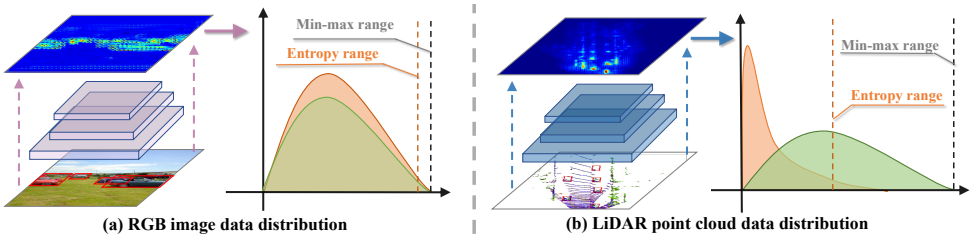
3.2 Sparsity-based Calibration

Here, in order to delve into the underlying reasons for the huge performance gap (31.29 mAPH/L2 in Tab 1) between Max-min and entropy calibrator. We statistically analyze the numerical distribution of feature maps of both RGB-based models and LiDAR-based object detection models, and visualize the main diversity as shown in Fig 3. The main reasons affecting the quantization performance can be summarized in two points:

Algorithm 1 LiDAR-PTQ quantization

Input: Pretrained FP model with N layers; Calibration dataset D^c, iteration T.
Output: quantization parameters of both activation and weight in network, i.e., weight scale s_w, weight zero-point z_w, activation scale s_a, activation zero-point z_a and adaptive rounding value for weight θ.

Optimize only weight quantization parameters s_w and z_w to minimize Eq 7 in every layer using the grid search algorithm;
input D^c to FP network to get the FP final output O_{fp}
for $L_n = \{L_i | i = 1, 2, ...N\}$ **do**
  Optimize only activation quantization parameters s_a and z_a to minimize Eq 7 in layer L_i using the grid search algorithm;
  Collect input data I_i to the FP layer L_i;
  Input I_i to quantized layer L_i^q and FP layer L_i to get quantized output $\hat{A}_i$ and FP output A_i;
  Input $\hat{A}_i$ to the following FP network to get output $\hat{O_{par}}$ of partial-quantized network;
  for all $j = 1, 2, \ldots, T$-iteration **do**
    Check quantized output $\hat{A}_i$ and FP output and calculate L_{local} using Eq 11;
    Check partial-quantized network output $\hat{O_{par}}$ and FP final output O_{fp} to calculate L_{TGPL} using Eq 9;
    Optimize quantization parameters s_w, z_w, s_a, and z_a, θ of layer L_i to minimize L_{total} using Eq 12;
  end for
  Quantize layer L_i with the learnable quantization parameters s_w, z_w, s_a, and z_a, θ;
end for



(a) RGB image data distribution (b) LiDAR point cloud data distribution

Figure 3: The diagram of data distribution for RGB-based and LiDAR-based object detection. Orange and green denote the data distribution of the entire feature map and foreground feature.

i) Huge sparsity lead to inappropriate quantization range. As shown in Fig 1 and 3, the sparsity of point cloud makes the whole BEV feature map exist a large number of zero pixels. Therefore, the entropy calibrator will statistic the feature value including zero pixels ($\approx 90\%$) to minimize the information loss, which leads to the values outside the quantization range being clipped. However, these truncated values contain rich geometric representations that could used for final object detection.

ii) Point cloud features are more sensitive to quantization range. Point cloud explicitly gauges spatial distances, and shapes of objects by collecting laser measurement signals from environments. During voxelization process, the raw point cloud coordinates, i.e., x, y, z in the ego-vehicle coordinate system are encoded as part of voxel features that preserve essential geometric information. Specifically, the arithmetic range of the input point cloud coordinates increases with detection distance. Therefore, the arithmetic range in the voxel feature is strongly correlated with detection distance. In other words, the arithmetic range of point cloud is relevant to the geometrics.

Furthermore, we also conduct an ablation study with different range distances on waymo val set. As shown in Tab 2, we find that *the decline in accuracy is exacerbated as the distance increases*. For entropy calibrator, the quantized performance on long-range metrics (50m - inf) is terribly damaged (5.90 mAPH/L2, up to 84.5% drop), while accuracy on short-range metrics (0-30m) remains well (60.03 mAPH/L2, 32.4% drop). This is because the entropy calibrator

Table 2: Ablation study in different range .

Method	Bits(W/A)	Vehicle LEVEL_2 mAPH		
		Mean 0-30 m	30-50 m	50-∞ m
Full Prec.	32/32	88.77	65.23	38.09
Entropy	8/8	60.03 **(-28.74)** (-32.4%)	22.46 **(-42.77)** (-65.6%)	5.90 **(-32.19)** (-84.5%)
Max-min	8/8	87.14 **(-1.63)** (-1.8%)	57.49 **(-7.74)** (-11.9%)	22.98 **(-15.11)** (-39.7%)

provides an inappropriate quantization range, resulting in a significant clipping error. Therefore, a large number of values with geometrics info are truncated and consequently a substantial degradation in model accuracy. On the contrary, for the Max-min calibrator, which covers the whole dynamic range of the FP activation, the values with geometric information are preserved effectively. Therefore, its performance in different range metrics performs well, especially on short-range metrics (0-30m), which only drops 1.63 mAPH/L2 (1.8%) than FP model.

Drawing upon the above findings, we conclude that the commonly used calibration method for RGB images is sub-optimal, while Max-min is more suitable for 3D point clouds. Therefore, we adopt Max-min calibrator for both weights and activations to mitigate the impact of high sparsity. Besides, to get more finer-grained scale factor and avoid the influence of outliers on rounding error Δr, a lightweight grid search (Banner et al., 2019; Choukroun et al., 2019) is incorporated to further optimize the quantization parameters.

Specifically, for a weight or activation tensor X, firstly obtain the x_{max} and x_{min} according to Eq.4, and calculate the initial quantization parameter s_0 following Eq. 3. Then linearly divide the interval $[\alpha s_0, \beta s_0]$ into T candidate bins, denoted as $\{s_t\}_{t=1}^{T}$. α, β and T are designed to control the search range and granularity. Finally, search $\{s_t\}_{t=1}^{T}$ to find the optimal s_{opt} that minimizes the quantization error,

$$\arg\min_{s_t} \|(X - \hat{X}(s_l))\|_F^2 \tag{7}$$

$\|\cdot\|_F^2$ is the Frobenius norm (MSE Loss). Refer to appendix for more details about grid search.

3.3 Task-Guided Global Positive Loss

The aforementioned calibration initialization approach can effectively improve the quantization accuracy of lidar detectors, but there is still a significant gap compared with the float model.

Both empirical and theoretical evidence (Li et al., 2021; Wei et al., 2022; Liu et al., 2023a) suggest that solely minimizing the quantization error in parameter space dose not guarantee equivalent minimization in final task loss within model space. Therefore, it becomes imperative to devise a global supervisory signal specifically tailored for 3D LiDAR-based detection tasks. This supervision would enable further fine-tuning of the quantization parameters p to achieve higher quantized precision. It is crucial to emphasize that this fine-tuning process does not involve labeled training data. Only need to minimize the distance between float output O^f and the quantized model's output $\hat{O}$, as depicted in Eq 8

$$\arg\min_{p}(O^f - \hat{O}) \tag{8}$$

In this paper, we propose Task-guided Global Positive Loss (TGPL) function to constrain the output disparity between the quantized and FP models. Our TGPL function features two characteristics that contribute to improving the performance of the quantized method:

i) Optimal quantization parameter on model space. The TGPL function compares the final output difference between the FP and the quantized models rather than the difference in each layer's output.

ii) Task-guided. As mentioned in Sec1 and Fig 1, there exists extreme imbalance between small informative foreground instances and large redundant background areas in Lidar-based detection tasks. For sparse 3D scenes, it is sub-optimal to imitate all feature pixels on dense 2D images. TGPL function is designed to leverage cues in the FP model's classification response to guide the quantized model to focus on the important area (*i.e.* positive sample location) that is relevant to final tasks.

In detail, we filter all prediction boxes from the FP model by a threshold γ, then we select the top K boxes. Then we perform NMS (Neubeck & Van Gool, 2006) to get the final prediction as positive boxes (pusedo-labels). Specifically, inspired by the Gaussian label assignment in CenterPoint (Yin et al., 2021), we define positive positions in a soft way with center-peak Gaussian distribution. Finally, for the classification branch, we use the focal loss (Lin et al., 2017) as the heatmap loss $\mathcal{L}_{cls}$. For the 3D box regression, we make use of the L1 loss $\mathcal{L}_{reg}$ to supervise their localization offsets, size and orientation. The overall TGPL loss consists of two parts as follows:

$$\mathcal{L}_{TGPL} = \mathcal{L}_{cls} + \alpha\mathcal{L}_{reg}, \tag{9}$$

3.4 ADAPTIVE ROUNDING-TO-NEAREST

Through grid search initialization and TGPL function constrain, the performance of quantized model has been greatly improved, but there is still a gap in achieving comparable accuracy with FP model. Recently, some methods (Wei et al., 2022; Liu et al., 2023a) optimize a variable, called rounding values, to determine whether weight values will be rounded up or down during the quantization process. In this way, the Eq 1 in weight quantization can be formulated as follows:

$$x_{int} = clamp(\lfloor \frac{x+\theta}{s} \rceil + z, q_{min}, q_{max}), \tag{10}$$

where θ is the optimization variable for each weight value to decide rounding results up or down (Nagel et al., 2020), *i.e.*, $\frac{\theta}{S}$ ranges from 0 to 1. Inspired by AdaRound(Nagel et al., 2020), we add a local reconstruction item to help learn the rounding value θ. The local reconstruction item as follows:

$$L_{Local} = \|(W_i \circledast I_i - \hat{W}_i \circledast I_i)\|_F^2 \tag{11}$$

where $\|\cdot\|_F^2$ is the Frobenius norm and $\hat{W}_i$ are the soft-quantized weights are calculated by Eq 10 and Eq 2. This operation allows us to adapt the rounding value to minimize information loss according to the calibration data, ensuring that the quantization process preserves important details. By adjusting the rounding value, we can achieve better performance of LiDAR-PTQ. Finally, the overall loss of our LiDAR-PTQ consists of two parts as follows:

$$\mathcal{L}_{total} = \lambda_1 \mathcal{L}_{local} + \lambda_2 \mathcal{L}_{TGPL}, \tag{12}$$

4 EXPERIMENTS

Dataset. To evaluate the effectiveness of our proposed Lidar-PTQ, we conduct main experiments on large-scale autonomous driving datasets, Waymo Open Dataset (WOD) (Sun et al., 2020).

Implementation Details. In WOD dataset, we randomly sample 256 frames point cloud data from the training set as the calibration data. The calibration set proportions is **0.16%** (256/158,081) for WOD. We set the first and the last layer of the network to keep full precision. The learning rate for the activation quantization scaling factor is 5e-5, and for weight quantization rounding is 5e-3. In TGPL loss, we set γ as 0.1, and K as 500. More details in supplements.

4.1 PERFORMANCE COMPARISON ON WAYMO DATASET

Table 3: Performance comparison on Waymo val set. ‡: reimplementation by us.

Models	Methods	Bits(W/A)	Mean (L2)		Vehicle (L2)		Pedestrian (L2)		Cyclist (L2)	
			mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH
CP-Pillar	Full Prec.	32/32	65.78	60.32	65.92	65.42	65.65	55.23	-	-
	BRECQ	8/8	61.73	56.27	61.87	61.36	61.59	51.18	-	-
	QDROP	8/8	63.60	58.14	63.74	63.23	63.46	53.04	-	-
	PD-QUANT	8/8	64.59	59.06	64.87	64.21	64.32	53.91	-	-
	QAT‡	8/8	65.56	60.08	65.69	65.17	65.44	54.99	-	-
	LiDAR-PTQ	8/8	**65.60**	**60.12**	**65.64**	**65.14**	**65.55**	**55.11**	-	-
CP-Voxel	Full Prec.	32/32	67.67	65.25	66.29	65.79	68.04	62.35	68.69	67.61
	BRECQ	8/8	63.15	60.71	62.53	62.03	63.22	57.49	63.71	62.60
	QDROP	8/8	64.70	62.23	63.97	63.38	64.90	59.17	65.24	64.13
	PD-QUANT	8/8	66.45	64.00	65.11	64.56	66.91	61.18	67.32	66.25
	QAT‡	8/8	67.63	65.20	66.28	65.76	67.98	62.28	68.62	67.55
	LiDAR-PTQ	8/8	**67.60**	**65.18**	**66.27**	**65.78**	**67.95**	**62.24**	**68.60**	**67.52**

Due to there are no PTQ methods specially designed for 3D LiDAR-based detection tasks, we reimplement several advanced PTQ methods in 2D RGB-based vision tasks, which are BRECQ (Li et al., 2021), QDROP (Wei et al., 2022) and PD-Quant (Liu et al., 2023a). Specifically, we select well-known CenterPoint (Yin et al., 2021) as our full precision model and report the quantized performance on WOD (Sun et al., 2020) dataset. Because it includes SPConv-based and SPConv-free models, which could effectively verify the generalization of our LiDAR-PTQ. As shown in Tab 3, LiDAR-PTQ achieves state-of-the-art performance and outperforms BRECQ and QDrop by a large margin of 3.87 and 2.00 on CenterPoint-Pillar model and 4.45 and 2.90 on CenterPoint-Voxel model.

For PD-Quant, a state-of-the-art PTQ method specially designed for RGB-based vision tasks, but it has suboptimal performance on LiDAR-based tasks. Specifically, to solve the over-fitting problem on the calibration set, PD-Quant adjusts activation according to FP model's BN layer. However, for the point cloud which is more sensitive to the arithmetic range, this design is ineffective and time-consuming, and will lead to accuracy loss. Notably, our LiDAR-PTQ achieves on-par or even superior accuracy than the QAT model and almost without performance drop than the float model.

4.2 The effectiveness of LiDAR-PTQ for Fully Sparse Detector

Table 4: The performance of FSD on Waymo val set.

Models	Methods	Bits(W/A)	Mean (L2)		Vehicle (L2)		Pedestrian (L2)		Cyclist (L2)	
			mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH
	Full Prec.	32/32	73.01	70.94	70.34	69.98	73.95	69.16	74.75	73.69
FSD	Entropy	8/8	10.54	9.44	0.06	0.06	21.88	18.86	9.69	9.41
	Max-min	8/8	71.24	69.37	68.42	68.18	72.09	67.60	73.22	72.34
	LiDAR-PTQ	8/8	**72.84**	**70.73**	**69.95**	**69.62**	**73.85**	**68.95**	**74.71**	**73.63**

Recently, there are emerging of some fully sparse 3D detectors, like FSD (Fan et al., 2022), FSD++ (Fan et al., 2023) and VoxelNext (Chen et al., 2023), etc. Here, we take FSD as an example to validate the effectiveness of our LiDAR-PTQ on fully sparse detectors. As shown in Tab 4, adopting entropy calibration still leads to a significant accuracy drop of **-61.50**. We discover that quantized FSD readily delivers the desired performance while employing a vanilla max-min calibration. Nonetheless, using LiDAR-PTQ can further achieve comparable accuracy to its float counterpart. The experiments demonstrate that LiDAR-PTQ is also applicable to fully sparse detectors.

4.3 Ablation Study

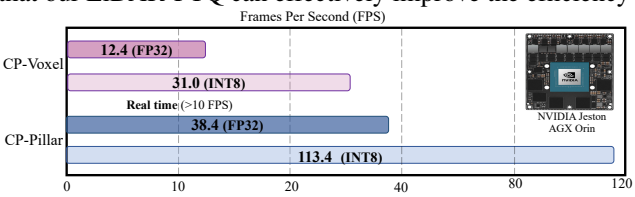
Table 5: Ablation study of different components of LiDAR-PTQ on Waymo val set.

Models	Methods	Bits(W/A)	Mean (L2)		Vehicle (L2)		Pedestrian (L2)	
			mAP	mAPH	mAP	mAPH	mAP	mAPH
	Full Prec.	32/32	65.78	60.32	65.92	65.42	65.65	55.23
CP-Pillar	Max-min	8/8	57.33	52.91	55.64	55.37	59.02	50.45
	+Grid S	8/8	63.66	58.39	63.37	62.87	63.96	53.91
	+TGPL	8/8	64.81	59.40	65.12	64.53	64.50	54.27
	+Round	8/8	**65.60**	**60.12**	**65.64**	**65.14**	**65.55**	**55.11**

Here, we conduct ablation study of different components in our LiDAR-PTQ based on CenterPoint-Pillar model to verify their effects. As shown in Tab 5, based on the selected Max-min calibrator, we could obtain 5.48 mAPH/L2 performance gain by using a lightweight grid search method. However, grid search only minimizes reconstruction error in parameter space, which is not equivalent to minimize the final performance loss. Therefore, by introducing the proposed TGPL function to fine-tune quantization parameters in model space, the performance of quantized model coud be 59.40 mAPH/L2. Finally, by introducing an adaptive rounding value, a freedom degree (Eq 10) is added to mitigate the final performance gap and achieve almost the same performance as the FP model (60.12 vs 60.32). Notably, the performance of FP model is the upper limit of the quantized model because we focus on post-training quantization without labeled training data.

4.4 Inference Acceleration

Here, we compared the speed of CenterPoint before and after quantization on an NVIDIA Jeston AGX Orin. This is a resource-constrained edge GPU platform that is widely used in real-word self-driving cars. The speed of quantized model enjoying $3\times$ inference speedup, which demonstrates that our LiDAR-PTQ can effectively improve the efficiency of 3D detection model on edge devices.



Method	Bits(W/A)	Latency	FPS
CP-Pillar	32/32	26.01	38.4
	8/8	8.82	113.4
CP-Voxel	32/32	80.64	12.4
	8/8	32.26	31.0

4.5 Computation Efficiency

LiDAR-PTQ requires additional computation and fine-tuning process compared to other traditional PTQ methods, resulting in increased time costs. While the quantization time is a limitation of LiDAR-PTQ, compared with other advanced PTQ methods, LiDAR-PTQ's additional time cost is acceptable. Furthermore, QAT method, the quantization time of LiDAR-PTQ is very short. For example, CenterPoint-Pillar will take 94 GPU/hour to achieve the same performance as the FP model on WOD dataset, while LiDAR-PTQ takes only 3 GPU/hour, which is 30 $\times$ faster than the QAT method. It also proves that our LiDAR-PTQ is cost-effective.

Model	QAT	BRECQ	QDROP	PD-QUANT	LiDAR-PTQ
CP-Pillar	93.82	1.93	1.82	6.72	2.75
CP-Voxel	80.51	1.73	1.64	6.13	2.12

5 Related Works

Post-training Quantization (PTQ). As mentioned in (Krishnamoorthi, 2018), existing quantization methods can be divided into two categories: (1) Quantization-Aware Training (QAT) and (2) Post-Training Quantization (PTQ). QAT methods (Wei et al., 2018; Li et al., 2019; Esser et al., 2019; Zhuang et al., 2020; Chen et al., 2021) require access to all labeled training data, which may not be feasible due to data privacy and security concerns. Compared to Quantization-aware Training (QAT) methods, Post-training quantization (PTQ) methods are simpler to use and allow for quantization with limited unlabeled data. Currently, there are many methods (Wu et al., 2020; Nahshan et al., 2021; Yuan et al., 2022; Liu et al., 2023b; Chu et al., 2024) designed for 2D vision tasks. AdaRound (Nagel et al., 2020) formulates the rounding task as a layer-wise quadratic unconstrained binary optimization problem and achieves a better performance. Based on AdaRound, BRECQ (Li et al., 2021) proposes utilizing block reconstruction to further enhance the accuracy of post-training quantization (PTQ). After that, QDrop (Wei et al., 2022) randomly drops the quantization of activations during PTQ and achieves new state-of-the-art accuracy. PD-Quant (Liu et al., 2023a) considers the global difference of models before and after quantization and adjusts the distribution of activation by BN layer statistics to alleviate the overfitting problem. However, these methods are specially designed for RGB images, and they are not readily transferable to LiDAR point cloud with substantial modal differences.

Quantization for 3D Object Detection. With the wide application of 3D object detection, in autonomous driving and robotics, some quantization methods are designed to improve inference speed for onboard deployment applications. With the advance of quantization techniques based on RGB image, QD-BEV (Zhang et al., 2023) achieves smaller size and faster speed than baseline BevFormer (Li et al., 2022) using QAT and distillation on multi-camera 3D detection tasks. For LiDAR-based 3D detection, especially for fully convolutional methods, like PointPillars (Lang et al., 2019), FCOS-LIDAR (Tian et al., 2022), FastPillars (Zhou et al., 2023), etc., effective quantization solutions could significantly speedup their latency to meet the practical requirements. (Stäcker et al., 2021) find that directly using INT8 quantization for 2D CNN will bring significant performance drop on PointPillars (Lang et al., 2019), where the reduction is even more severe for the entropy calibrator. Besides, BiPointNet (Qin et al., 2021) is a binarization quantization method, which focuses on classification and segmentation tasks based on point cloud captured from small CAD simulation. To the best of our knowledge, there is no quantization solution designed for large-scale outdoor LiDAR-based 3D object detection methods in self-driving.

6 Conclusion and Future Work

In this paper, we analyze the root cause of the performance degradation of point cloud data during the quantization process. Then we propose an effective PTQ method called LiDAR-PTQ, which is particularly designed for 3D LiDAR-based object detection tasks. Our LiDAR-PTQ features three main components: **(1)** a sparsity-based calibration method to determine the initialization of quantization parameters **(2)** a Task-guided Global Positive Loss (TGPL) to reduce the disparity on the final task. **(3)**. an adaptive rounding-to-nearest operation to minimize the layer-wise reconstruction error. Extensive experiments demonstrate that our LiDAR-PTQ can achieve state-of-the-art performance on CenterPoint (both pillar-based and voxel-based). To our knowledge, for the very first time in lidar-based 3D detection tasks, the PTQ INT8 model's accuracy is almost the same as the FP32 model while enjoying $3\times$ inference speedup. Moreover, our LiDAR-PTQ is cost-effective being $30\times$ faster than the quantization-aware training method. Given its effectiveness and efficiency, we hope that our LiDAR-PTQ can serve as a valuable quantization tool for current mainstream grid-based 3D detectors and push the development of practical deployment of 3D detection models on edge devices. Besides, we believe that the low-bit quantization of 3D detectors will bring further efficiency improvement. This remains an open problem for future research.

ACKNOWLEDGMENTS

We thank anonymous reviewers for their kind help of this work. This work was supported by National Key R&D Program of China (No. 2022ZD0118700), National Natural Science Foundation of China (No.62271143), and the Big Data Computing Center of Southeast University.

REFERENCES

Ron Banner, Yury Nahshan, and Daniel Soudry. Post training 4-bit quantization of convolutional networks for rapid-deployment. *Advances in Neural Information Processing Systems*, 32, 2019. 3, 6

Peng Chen, Jing Liu, Bohan Zhuang, Mingkui Tan, and Chunhua Shen. Aqd: Towards accurate quantized object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 104–113, 2021. 9

Yukang Chen, Jianhui Liu, Xiangyu Zhang, Xiaojuan Qi, and Jiaya Jia. Voxelnext: Fully sparse voxelnet for 3d object detection and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 8

Yoni Choukroun, Eli Kravchik, Fan Yang, and Pavel Kisilev. Low-bit quantization of neural networks for efficient inference. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 3009–3018. IEEE, 2019. 6

Xiangxiang Chu, Liang Li, and Bo Zhang. Make repvgg greater again: A quantization-aware approach. *AAAI*, 2024. 9

Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. Learned step size quantization. *arXiv preprint arXiv:1902.08153*, 2019. 9

Lue Fan, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Fully sparse 3d object detection. *Advances in Neural Information Processing Systems*, 35:351–363, 2022. 8

Lue Fan, Yuxue Yang, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Super sparse 3d object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 8

Yongkweon Jeon, Chungman Lee, Eulrang Cho, and Yeonju Ro. Mr. biq: Post-training non-uniform quantization based on minimizing the reconstruction error. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12329–12338, 2022. 3

Xiaogang Wang Hongsheng Li Jiageng Mao, Shaoshuai Shi. 3d object detection for autonomous driving: A comprehensive survey. *IJCV*, 2023. 1

Raghuraman Krishnamoorthi. Quantizing deep convolutional networks for efficient inference: A whitepaper. *arXiv preprint arXiv:1806.08342*, 2018. 9

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*, pp. 12697–12705, 2019. 9

Rundong Li, Yan Wang, Feng Liang, Hongwei Qin, Junjie Yan, and Rui Fan. Fully quantized network for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2810–2819, 2019. 9

Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. Brecq: Pushing the limit of post-training quantization by block reconstruction. 2021. 1, 6, 7, 9

Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX*, pp. 1–18. Springer, 2022. 9

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017. 6

Jiawei Liu, Lin Niu, Zhihang Yuan, Dawei Yang, Xinggang Wang, and Wenyu Liu. Pd-quant: Post-training quantization based on prediction difference metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24427–24437, 2023a. 6, 7, 9

Yijiang Liu, Huanrui Yang, Zhen Dong, Kurt Keutzer, Li Du, and Shanghang Zhang. Noisyquant: Noisy bias-enhanced post-training activation quantization for vision transformers. 2023b. 9

Szymon Migacz. 8-bit inference with tensorrt. In *GPU technology conference*, volume 2, pp. 5, 2017. 3

Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *International Conference on Machine Learning*, pp. 7197–7206. PMLR, 2020. 1, 7, 9

Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart Van Baalen, and Tijmen Blankevoort. A white paper on neural network quantization. *arXiv preprint arXiv:2106.08295*, 2021. 3, 4

Yury Nahshan, Brian Chmiel, Chaim Baskin, Evgenii Zheltonozhskii, Ron Banner, Alex M Bronstein, and Avi Mendelson. Loss aware post-training quantization. volume 110, pp. 3245–3262. Springer, 2021. 9

Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 3, pp. 850–855. IEEE, 2006. 6

Haotong Qin, Zhongang Cai, Mingyuan Zhang, Yifu Ding, Haiyu Zhao, Shuai Yi, Xianglong Liu, and Hao Su. Bipointnet: Binary neural network for point clouds. In *ICLR*, 2021. 9

Lukas Stäcker, Juncong Fei, Philipp Heidenreich, Frank Bonarens, Jason Rambach, Didier Stricker, and Christoph Stiller. Deployment of deep neural networks for object detection on edge ai devices with runtime optimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshop*, pp. 1015–1022, 2021. 4, 9

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. pp. 2446–2454, 2020. 2, 7

Zhi Tian, Xiangxiang Chu, Xiaoming Wang, Xiaolin Wei, and Chunhua Shen. Fully convolutional one-stage 3d object detection on lidar range images. *Advances in Neural Information Processing Systems*, 35:34899–34911, 2022. 9

Xiuying Wei, Ruihao Gong, Yuhang Li, Xianglong Liu, and Fengwei Yu. Qdrop: randomly dropping quantization for extremely low-bit post-training quantization. 2022. 1, 6, 7, 9

Yi Wei, Xinyu Pan, Hongwei Qin, Wanli Ouyang, and Junjie Yan. Quantization mimic: Towards very tiny cnn for object detection. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 267–283, 2018. 9

Di Wu, Qi Tang, Yongle Zhao, Ming Zhang, Ying Fu, and Debing Zhang. Easyquant: Post-training quantization via scale optimization. *arXiv preprint arXiv:2006.16669*, 2020. 9

Hongyi Yao, Pu Li, Jian Cao, Xiangcheng Liu, Chenying Xie, and Bingzhang Wang. Rapq: Rescuing accuracy for power-of-two low-bit post-training quantization. *arXiv preprint arXiv:2204.12322*, 2022. 1

Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. pp. 11784–11793, 2021. 2, 4, 6, 7

Zhihang Yuan, Chenhao Xue, Yiqi Chen, Qiang Wu, and Guangyu Sun. Ptq4vit: Post-training quantization for vision transformers with twin uniform quantization. pp. 191–207, 2022. 9

Yifan Zhang, Zhen Dong, Huanrui Yang, Ming Lu, Cheng-Ching Tseng, Yandong Guo, Kurt Keutzer, Li Du, and Shanghang Zhang. Qd-bev: Quantization-aware view-guided distillation for multi-view 3d object detection. 2023. 9

Sifan Zhou, Zhi Tian, Xiangxiang Chu, Xinyu Zhang, Bo Zhang, Xiaobo Lu, Chengjian Feng, Zequn Jie, Patrick Yin Chiang, and Lin Ma. Fastpillars: A deployment-friendly pillar-based 3d detector. *arXiv preprint arXiv:2302.02367*, 2023. 9

Bohan Zhuang, Lingqiao Liu, Mingkui Tan, Chunhua Shen, and Ian Reid. Training quantized neural networks with a full-precision auxiliary module. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1488–1497, 2020. 9