3rd Place Solution of Waymo Open Dataset Challenge 2021 Real-time 3D Detection Track

Han Liu, Rongzhang Zheng, Jinzhang Peng, Lu Tian
Xilinx Inc., Beijing, China.
{hanl, treemann, jinzhang, lutian}@xilinx.com

Abstract

This technical report introduces details about our method for the Waymo Open Dataset (WOD) Challenge 2021 Real-time 3D Detection Track. CenterPoint is chosen as our baseline model for its simplicity and effectiveness. To improve the accuracy, we merge multi-sweep LiDAR point clouds as input, increase the spatial resolution for voxelization, apply the structural re-parameterization technique to the backbone, adopt a two-stage framework and use Quality Focal Loss for training. We also conduct multiple optimization strategies to reduce the inference time, including converting the Pytorch model to TensorRT, deploying with half-precision and accelerating data preparation, voxelization and the points-in-proposals process with CUDA. Our final submission achieves 70.46% *mAPH/L2 with 68.4ms, striking a balance between accuracy and speed.*

1. Datasets

The Waymo Open Dataset [13] is the largest and most diverse multimodal autonomous driving dataset up to now. It contains images from multiple high-resolution cameras and point clouds from multiple high-quality LiDAR sensors, with 12 million LiDAR box annotations and around 12 million camera box annotations. It consists of 1000 scenes for training and validation, and 150 scenes for testing, where each scene spans 20s. For the Real-time 3D Detection Track, a set of 3D upright boxes should be predicted for each frame, given the LiDAR point clouds and the camera images. The metric for ranking is mAPH/L2, which stands for the mean over the Average Precision with Heading (APH) of Vehicles, Cyclists, and Pedestrians at difficulty of LEVEL_2. The award of the challenge takes both latency and accuracy into consideration.

2. Methods

In the autonomous driving scene, objects of different classes have various sizes and orientations, which increases the complexity of anchor-based 3D object detectors' design, so we choose the state-of-the-art anchor-free detector CenterPoint [14] as our baseline model. Based on CenterPoint, we adopt multiple optimization strategies to improve the performance and reduce the inference time.

2.1. Baseline model

Adopting the idea of the 2D image-based detection model CenterNet [15], CenterPoint simplifies the 3D object location task as object center detection with heatmap and regresses 3D size and orientation of the detected objects. Compared with anchor-based methods, CenterPoint achieves higher performance with less complexity. The framework of CenterPoint is compatible with off-the-shelf voxel-based or pillar-based 3D object detectors, making it flexible to scale the model and strike a balance between speed and accuracy. We use CenterPoint-Pillar as our baseline model.

We utilize the following data augmentation strategies for training. First we use the effective crop and paste augmentation. Specifically, during data preparation, we generate a database containing labels and the corresponding point cloud data. Then 15, 10 and 10 ground-truth objects (with not less than 5 points in the 3D bounding box) are randomly selected for the vehicle, pedestrian and cyclist, respectively, and pasted to the current frame. In addition, we conduct global rotation following uniform distribution $U(-\pi/4, \pi/4)$ and global scaling following uniform distribution $U(0.95, 1.05)$ to make the model more robust to noise disturbance.

We adopt AdamW optimizer [9] with betas of $(0.9, 0.99)$, weight decay of 0.01 and do not use the AMSGrad variant for training. We use the one-cycle learning rate scheduler with the max learning rate of 3×10^{-3}, division factor of 10 and cyclic momentum of $(0.95, 0.85)$. Gradient clipping (i.e., forcing the gradient to 35 if its L2-norm exceeds 35) is also used to make the loss convergence more stable. We train the model on the training set for 36 epochs with 8 samples per batch on V100 GPU.

For inference, we first use a score threshold of 0.1 to fil-

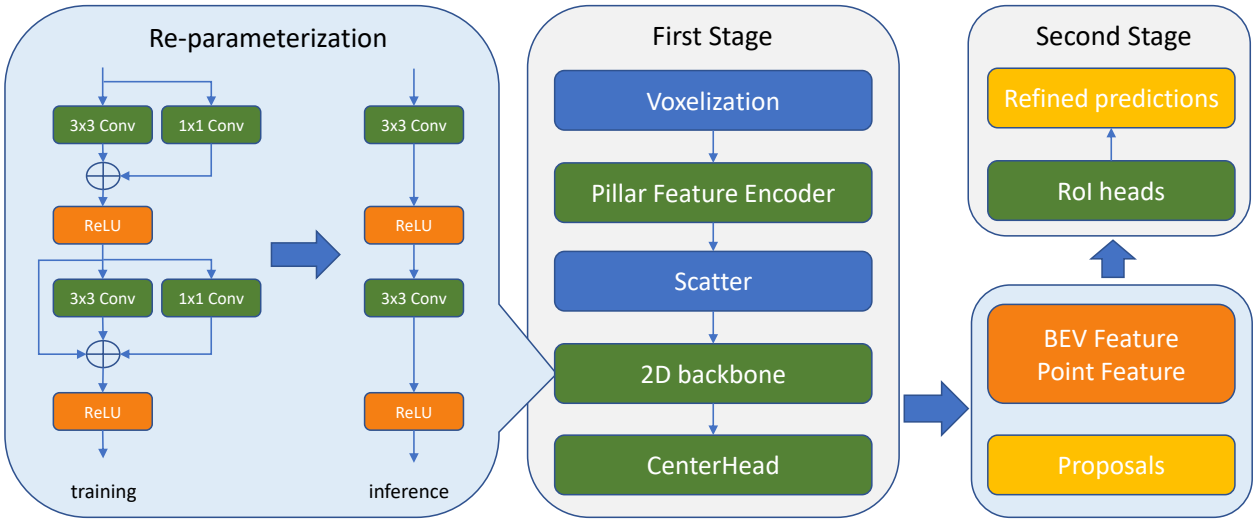

Figure 1. Our algorithm framework for 3D object detection.

ter boxes, and then apply Rotate Non-Maximum Suppression with the IoU threshold of 0.7 to get final top 500 predictions. The baseline achieves 60.49% mAPH at LEVEL 2.

2.2. Bells and whistles

In this section, we show improvements of different algorithmic components used in our model.

Structural Re-Parameterization. We adopt structural re-parameterization technique [2] to boost the accuracy without extra cost on the latency. Specifically, as shown in Figure 1, we use multi-branch architecture by adding identity and 1x1 convolution connection during training. At inference time, we merge these branches to construct a plain architecture for the backbone, which is the same as the baseline model. Thus, we can maintain the same latency cost but get higher performance. As shown in Table 1, model with structural re-parameterization achieves 61.38% mAPH at LEVEL 2 among ALL_NS, outperforming the baseline by 0.89%.

Methods	Vehicle	Pedestrian	Cyclist	ALL_NS
Baseline	65.31	55.51	60.66	60.49
REP	66.41	57.58	60.14	61.38

Table 1. 3D detection results in terms of APH/L2 (%) on WOD validation set. REP stands for baseline model with structural re-parameterized backbone.

Merging Multi-Sweep. Due to the sparsity of point cloud, we accumulate LiDAR sweeps to densify the LiDAR point cloud as in [1]. In the Waymo challenge, we use the past frame combined with the current frame as our input point cloud. To distinguish points from two sweeps, pseudo time difference Δt is attached to the point cloud. $\Delta t = 0$ indicates point cloud at current frame and $\Delta t = 1$ indicates point cloud at past frame. We treat pseudo time difference Δt as an additional attribute and combine current frame and past frame point cloud as the input in our experiment. The input format should be $(x, y, z, intensity, elongation, \Delta t)$. As shown in Table 2, this method improves the accuracy by 4.95%.

Methods	Vehicle	Pedestrian	Cyclist	ALL_NS
REP	66.41	57.58	60.14	61.38
REP+Two Sweeps	68.36	64.37	66.27	66.33

Table 2. 3D detection results in terms of APH/L2 (%) on WOD validation set. REP stands for baseline model with structural re-parameterized backbone.

Higher Spatial Resolution. As mentioned in [5], we can change the size of the spatial binning to get a trade-off between speed and accuracy. We change the grid size of (0.32m, 0.32m) used in CenterPoint-Pillar [14] to the size of (0.24m, 0.24m), which enhances performance while keeping latency cost in an acceptable range. We report results of different grid sizes in Table 3.

Two Stage Head. A second stage process is designed to refine the predictions from CenterPoint. Inspired by [8, 11], we deduce that combining raw point features can compensate for information loss by voxel-based methods. We enlarge the proposals from the first stage by 1.5m each side to include contextual points. Then 512 points are sampled for each enlarged box and normalized according to the proposal box coordinate system. If the number of points in one proposal box is fewer than required, we repeat the sampled points to keep the input dimension. To fix the size ambiguity problem, we also add boundary off-

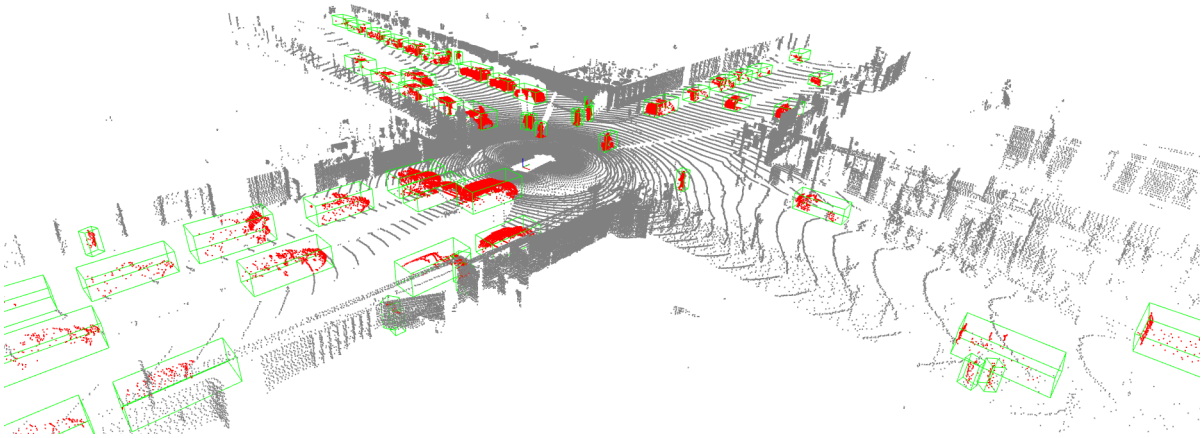

Figure 2. Examples of our detection results on the WOD test set.

Methods	Vehicle	Pedestrian	Cyclist	ALL_NS	Latency ms
REP 0.32m	66.41	57.58	60.14	61.38	63.8
REP 0.28m	67.01	61.02	63.48	63.84	73.2
REP 0.24m	67.54	61.33	64.01	64.29	86.0

Table 3. 3D detection results in terms of APH/L2 (%) on WOD validation set. We report the results of different grid sizes. REP stands for baseline model with structural re-parameterized backbone.

set information to each point following [8]. We calculate boundary offset in the proposal box coordinate system for each point and append to point features. Each point now has $D = 12$ dimensions. The format of point should be $(x, y, z, intensity, elongation, \Delta t, [...offset...])$. We use a point-based feature extractor which is composed of a Batch Normalization [3] and a simplified version of PointNet [5]. We stack sampled points in each proposal to create a dense tensor of size (N, D, P) as an input feature, where N stands for the number of proposal boxes and P stands for sampled points per proposal. The point-based feature extractor map the original feature to feature embedding with 512 channels following [8]. The size of output features should be (N, C, P). C is outputted channel size of features. Then we use a max-pooling operation to compresses features to tensor of size (N, C). Finally, we concatenate point-based features of each proposal box to its corresponding grid-based features as the input features of the prediction head. If the proposal box contains no points, we pad the point-based feature with 0 and mark its ground-truth class as background. Table 4 shows that the two-stage model with point-based features outperforms the model without point-based features by 1.19%. For class-agnostic confidence score prediction, [14] uses a score guided by the box's 3D IoU as target [4, 6, 11, 12]. Differently, we directly use the box's 3D IoU as our target and optimize the confidence score using Quality Focal Loss (QFL) [7]:

$$QFL(\sigma) = -|y - \sigma|^{\beta} * [(1 - y)log(1 - \sigma) + log(\sigma)] \quad (1)$$

where σ is the predicted IoU, y is the ground truth and β is set to 2.0. Training with QFL can further increase the performance by 0.3% mAPH at LEVEL 2.

Methods	Vehicle	Pedestrian	Cyclist	ALL_NS
REP	66.41	57.58	60.14	61.38
REP +Second Stage	68.00	58.30	61.88	62.73
REP +Second Stage +Point-Based Feature	68.11	59.76	63.89	63.92

Table 4. 3D detection results in terms of APH/L2 (%) on WOD validation set. REP stands for baseline model with structural re-parameterized backbone.

Latency Optimization. To accelerate the model on Nvidia Tesla V100 GPU, we conduct latency optimization in three aspects to achieve the target of 70 ms/frame. First, at data pre-processing step, we need to process twenty sets of point cloud data every detection frame [1]. In order to utilize parallelization, we divide them to two sets by current frame and previous frame and use tensor CUDA stream API in PyTorch [10] to process them in parallel. Second, we implement pillar-generation step and points-in-proposals step using CUDA. Third, we convert our model to TensorRT and deploy it with half-precision for inference.

[1]For every detection frame, we should accumulate two sweeps of point cloud data, each of them contains five LiDAR sensors' points on both 1st and 2nd returns.

3. Final Results

For final submission, we adopt all the aforementioned strategies for performance improvement and latency reduction. In the first stage, we train the model with 21 samples per batch and triple the training cycle. In the second stage, we freeze the network at the first stage and train the model with 48 samples per batch and max learning rate 7.5×10^{-4}. Results are shown in Table 5.

Methods	Vehicle	Pedestrian	Cyclist	ALL_NS
Baseline	65.31	55.51	60.66	60.49
X_Autonomous3D(Ours)	73.60	68.27	69.50	70.46

Table 5. Submission results in terms of APH/L2 (%) on WOD testing set.

References

[1] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[2] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13733–13742, June 2021.

[3] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[4] Borui Jiang, Ruixuan Luo, Jiayuan Mao, Tete Xiao, and Yuning Jiang. Acquisition of localization confidence for accurate object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 784–799, 2018.

[5] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12697–12705, 2019.

[6] Buyu Li, Wanli Ouyang, Lu Sheng, Xingyu Zeng, and Xiaogang Wang. Gs3d: An efficient 3d object detection framework for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1019–1028, 2019.

[7] Xiang Li, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. In *NeurIPS*, 2020.

[8] Zhichao Li, Feng Wang, and Naiyan Wang. Lidar r-cnn: An efficient and universal 3d object detector. *CVPR*, 2021.

[9] Ilya Loshchilov and F. Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[10] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[11] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020.

[12] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[13] 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.

[14] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. *CVPR*, 2021.

[15] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. In *arXiv preprint arXiv:1904.07850*, 2019.