Supplemental Material

Anonymous Author(s)
Affiliation
Address
email

Our supplementary materials covers the following: background on 3D object detection in the range view, additional quantitative results, qualitiative results, dataset details, and implementation details for our models.

1 Range View Representation

The range view representation, also known as a range image, is a 2D grid containing the spherical coordinates of an observed point with respect to the lidar laser's original reference frame. We define a range image as:

$$r \triangleq \{(\varphi_{ij}, \theta_{ij}, r_{ij}) : 1 \leq i \leq H; 1 \leq j \leq W\}, \tag{1}$$

where $(\varphi_{ij}, \theta_{ij}, r_{ij})$ are the inclination, azimuth, and range, and H, W are the height and width of the image. Importantly, the cells of a range image are not limited to containing only spherical coordinates. They may also contain auxillary sensor information such as a lidar's intensity.

1.1 3D Object Detection

Given a range image r, we construct a set of 3D object proposals which are ranked by a confidence score. Each proposal consists of a proposed location, size, orientation, and category. Let $\mathcal{D}$ represent are predictions from a network.

$$\mathcal{D} \triangleq \left\{d_i \in \mathbb{R}^8\right\}_{i=1}^{K}, \text{ where } K \subset \mathbb{N}, \tag{2}$$

$$d_i \triangleq \left\{x_i^{\text{ego}}, y_i^{\text{ego}}, z_i^{\text{ego}}, l_i, w_i, h_i, \theta_i, c_i\right\} \tag{3}$$

where $x_i^{\text{ego}}, y_i^{\text{ego}}, z_i^{\text{ego}}$ are the coordinates of the object in the ego-vehicle reference frame, l_i, w_i, h_i are the length, width, and height of the object, θ_i is the counter-clockwise rotation about the vertical axis, and c_i is the object likelihood. Similarly, we define the ground truth cuboids as:

$$\mathcal{G} \triangleq \left\{g_i \in \mathbb{R}^8\right\}_{i=1}^{M}, \text{ where } M \subset \mathbb{N}, \tag{4}$$

$$g_i \triangleq \left\{x_i^{\text{ego}}, y_i^{\text{ego}}, z_i^{\text{ego}}, l_i, w_i, h_i, \theta_i, q_i\right\}, \tag{5}$$

where q_i is a continuous value computed dynamically during training. For example, q_i may be set to Dynamic 3D Centerness or IoU_{BEV}. The detected objects, $\mathcal{D}$ are decoded as the same parameterization as $\mathcal{G}$.

$$\mathcal{D} \triangleq \left\{d_k \in \mathbb{R}^8 : c_1 \geq \cdots \geq c_k\right\}_{k=1}^{K}, \text{ where } K \subset \mathbb{N}, \tag{6}$$

$$d_k \triangleq \left\{x_k^{\text{ego}}, y_k^{\text{ego}}, z_k^{\text{ego}}, l_k, w_k, h_k, \theta_k\right\}. \tag{7}$$

We seek to predict a continuous representation of the ground truth targets as:

$$\mathcal{D} \triangleq \left\{d_k \in \mathbb{R}^8 : c_1 \geq \cdots \geq c_k\right\}_{k=1}^{K}, \text{ where } K \subset \mathbb{N}, \tag{8}$$

$$g_k \triangleq \left\{x_k^{\text{ego}}, y_k^{\text{ego}}, z_k^{\text{ego}}, l_k, w_k, h_k, \theta_k, c_k\right\}, \tag{9}$$

where $x_k^{\text{ego}}, y_k^{\text{ego}}, z_k^{\text{ego}}$ are the coordinates of the object in the ego-vehicle reference frame, l_k, w_k, h_k are the length, width, and height of the object, θ_k is the counter-clockwise rotation about the vertical axis, and c_k is the object category likelihood.

3D Anchor Points in the Range View. To predict objects, we bias our predictions by the location of *observed* 3D points which are features of the projected pixels in a range image. For all the 3D points contained in a range image, we produce a detection d_k.

Regression Targets. Following previous literature, we do not directly predict the object proposal representation in Section 1.1. Instead, we define the regression targets as the following:

$$\mathcal{T}(\mathcal{P}, \mathcal{G}) = \{t_i(p_i, g_i) \in \mathbb{R}^8\}_{i=1}^{K}, \text{ where } K \in \mathbb{N}, \tag{10}$$

$$t_i(p_i, g_i) = \{\Delta x_i, \Delta y_i, \Delta z_i, \log l_i, \log w_i, \log h_i, \sin\theta_i, \cos\theta_i\}, \tag{11}$$

where $\mathcal{P}$ and $\mathcal{G}$ are the sets of points in the range image and the ground truth cuboids in the 3D scene, $\Delta x_i, \Delta y_i, \Delta z_i$ are the offsets from the point to the associated ground truth cuboid in the point-azimuth reference frame, $\log l_i, \log w_i, \log h_i$ are the logarithmic length, width, and height of the object, respectively, and $\sin\theta_i, \cos\theta_i$ are continuous representations of the object's heading θ_i.

Classification Loss. Once all of the candidate foreground points have been ranked and assigned, each point needs to incur loss proportional to its regression quality. We use Varifocal loss [1] with a sigmoid-logit activation for our classification loss:

$$\text{VFL}(c_i, q_i) = \begin{cases} q_i(-q_i \log(c_i) + (1-q_i)\log(1-c_i)) \text{ if } q_i > 0 \\ -\alpha c_i^{\gamma} \log(1-c_i) \text{ otherwise,} \end{cases} \tag{12}$$

where c_i is classification likelihood and q_i is 3D classification targets (*e.g.*, Dynamic IoU$_{\text{BEV}}$ or Dynamic 3D Centerness). Our final classification loss for an entire 3D scene is:

$$\mathcal{L}_c = \frac{1}{M} \sum_{j=1}^{N} \sum_{i=1}^{|\mathcal{P}_G^j|} \text{VFL}(c_i^j, q_i^j), \tag{13}$$

where M is the total number of foreground points, N is the total number of objects in a scene, $\mathcal{P}_G^j$ is the set of 3D points which fall inside the j^{th} ground truth cuboid, c_i^j is the likelihood from the network classification head, and q_i^j is the 3D classification target.

Regression Loss. We use an ℓ_1 regression loss to predict the regression residuals. The regression loss for an entire 3D scene is:

$$\mathcal{L}_r = \frac{1}{N} \sum_{j=1}^{N} \frac{1}{|\mathcal{P}_G^j|} \sum_{i=1}^{|\mathcal{P}_G^j|} \text{L1Loss}(r_i^j, t_i^j), \tag{14}$$

where N is the total number of objects in a scene, $\mathcal{P}_G^j$ is the set of 3D points which fall inside the j^{th} ground truth cuboid, r_i^j is the predicted cuboid parameters from the network, and t_i^j are the target residuals to be predicted.

Total Loss. Our final loss is written as:

$$\mathcal{L} = \mathcal{L}_c + \mathcal{L}_r \tag{15}$$

1.2 Argoverse 2

Additional details on the evaluation metrics used in the Argoverse 2.

- **Average Precision (AP)**: VOC-style computation with a true positive defined at 3D Euclidean distance averaged over $0.5\,\text{m}$, $1.0\,\text{m}$, $2.0\,\text{m}$, and $4.0\,\text{m}$.
- **Average Translation Error (ATE)**: 3D Euclidean distance for true positives at $2\,\text{m}$.
- **Average Scale Error (ASE)**: Pose-aligned 3D IoU for true positives at $2\,\text{m}$.
- **Average Orientation Error (AOE)**: Smallest yaw angle between the ground truth and prediction for true positives at $2\,\text{m}$.

- **Composite Detection Score (CDS)**: Weighted average between AP and the normalized true positive scores:

$$\text{CDS} = \text{AP} \cdot \sum_{x \in \mathcal{X}} 1 - x, \text{ where } x \in \{\text{ATE}_{\text{unit}}, \text{ASE}_{\text{unit}}, \text{AOE}_{\text{unit}}\}. \quad (16)$$

We refer readers to Wilson *et al.* [2] for further details.

1.3 Waymo Open

Additional details on the evaluation metrics used in the Waymo Open are listed below.

1. **3D Mean Average Precision (mAP)**: VOC-style computation with a true positive defined by 3D IoU. The gravity-aligned-axis is fixed.
   (a) **Level 1 (L1)**: All ground truth cuboids with at least five lidar points within them.
   (b) **Level 2 (L2)**: All ground cuboids with at least 1 point and additionally incorporates heading into its true positive criteria.

Following RangeDet [3], we report L1 results.

2 Range-view 3D Object Detection

Baseline Model. Our baseline models are all multi-class and utilize the Deep Layer Aggregation (DLA) [4] architecture with an input feature dimensionality of 64. In our Argoverse 2 experiments, we incorporate five input features: x, y, z, range, and intensity, while for our Waymo experiments, we include six input features: x, y, z, range, intensity, and elongation. These inputs are then transformed to the backbone feature dimensionality of 64 using a single basic block. For post-processing, we use weighted non-maximum suppression (WNMS). All models are trained and evaluated using mixed-precision with BrainFloat16 [5]. Both models use a OneCycle scheduler with AdamW using a learning rate of 0.03 across four A40 gpus. All models in the ablations are trained for 5 epochs on a uniformly sub-sampled fifth of the training set.

State-of-the-art Comparison Model. We leverage the best performing and most general methods from our experiments for our state-of-the-art comparison for both the Argoverse 2 and Waymo Open dataset models. The Argoverse 2 and Waymo Open models use an input feature dimensionality of 256 and 128, respectively. Both models uses the Meta-Kernel and a 3D input encoding, Dynamic 3D Centerness for their classification supervision, and we use our proposed Range-Subsampling with range partitions of [0 - 30 m), [30 m, 50 m), [50 m, ∞) with subsampling rates of 8, 2, 1, respectively. For both datasets, models are trained for 20 epochs.

Method	**mAP** ↑	**ATE** ↓	**ASE** ↓	**AOE** ↓	**CDS** ↑
Dynamic IoU_{BEV} [3]	14.2	0.87	0.51	1.24	10.9
Dynamic 3D Centerness (ours)	**16.9**	**0.77**	**0.46**	**1.04**	**12.8**

Table 1: **Classification Supervision: Argoverse 2.** Evaluation metrics and errors using two different classification supervision methods on the Argoverse 2 *validation* set. We observe that our Dynamic 3D Centerness method outperforms all methods. Surprisingly, Dynamic 3D centerness outperforms IoU_{BEV} in average translation, scale, orientation errors.

2.1 Qualitative Results

We include qualitative results for both Argoverse 2 and Waymo Open shown in Figs. 1 and 2.

Method	3D AP$_{L1}$ ↑		
	Vehicle	Pedestrian	Cyclist
Dynamic IoU$_{\text{BEV}}$ [3]	59.90	67.08	25.52
Dynamic 3D Centerness (ours)	**59.98**	**68.03**	**34.66**

Table 2: **Classification Supervision: Waymo Open.** Evaluation metrics and errors using two different classification supervision methods on the Waymo Open *validation* set. Our results suggest that Dynamic 3D Centerness is a competitive alternative to IoU$_{\text{BEV}}$, while being simpler.

Method	mAP ↑	ATE ↓	ASE ↓	AOE ↓	CDS ↑
Basic Block	16.7	**0.78**	**0.47**	**1.15**	12.7
Meta Kernel [3]	**18.7**	0.80	0.50	1.18	**14.1**
Range Aware Kernel⋆ [6]	16.3	0.81	0.51	1.23	12.4

Table 3: **3D Input Encoding: Argoverse 2.** Mean Average Precision using different 3D input feature encodings on the Argoverse 2 *validation* set. ⋆: Code unavailable. Re-implemented by ourselves.

Method	3D AP$_{L1}$ ↑		
	Vehicle	Pedestrian	Cyclist
Basic Block	60.27	66.95	22.42
Meta Kernel [3]	**64.44**	**72.75**	**43.52**
Range Aware Kernel⋆ [6]	60.00	66.42	18.54

Table 4: **3D Input Encoding: Waymo Open.** L1 Average Precision (AP) across three different 3D input feature encodings on the Waymo *validation* set. The Meta Kernel outperforms all methods improving AP considerably across all categories. Surprisingly, the Range Aware Kernel performs worse than our baseline method. ⋆: Code unavailable. Re-implemented by ourselves based on details in the manuscript [6].

	Mean	**R. Vehicle**	**Pedestrian**	**Bollard**	**C. Barrel**	**C. Cone**	**S. Sign**	**Bicycle**	**L. Vehicle**	**B. Truck**	**W. Device**	**Sign**	**Bus**	**V. Trailer**	**Truck**	**Motorcycle**	**T. Cab**	**Bicyclist**	**S. Bus**	**W. Rider**	**Motorcyclist**	**Dog**	**A. Bus**	**M.P.C. Sign**	**Stroller**	**Wheelchair**	**M.B. Trailer**
Distribution (%)	-	56.92	17.95	6.8	3.62	2.63	1.99	1.42	1.25	1.09	1.06	0.91	0.83	0.69	0.54	0.47	0.44	0.38	0.2	0.18	0.16	0.15	0.1	0.08	0.06	0.05	0.0
mAP ↑																											
CenterPoint [7]	22.0	67.6	46.5	40.1	32.2	29.5	-	24.5	3.9	37.4	-	6.3	38.9	22.4	22.6	33.4	-										
FSD [8]	28.2	68.1	59.0	41.8	64.9	41.2	-	38.6	5.9	38.5	-	11.9	40.9	**26.9**	14.8	49.0	-	33.4	30.5	-	39.7	-	**20.4**	26.4	13.8	-	-
VoxelNext [9]	30.7	72.7	63.2	**53.9**	64.9	44.9	-	**40.6**	6.8	**40.1**	-	14.9	38.8	20.9	19.9	42.4	-	32.4	25.2	-	**44.7**	-	20.1	39.4	15.7	-	-
Ours	**31.6**	**75.7**	**67.2**	48.6	**70.2**	**50.3**	39.6	40.0	**7.5**	32.6	19.6	**18.0**	**44.0**	22.0	**24.0**	**49.5**	19.3	**34.4**	**44.0**	5.5	42.1	5.8	9.9	**41.5**	**17.7**	3.4	0.0

Table 5: **State-of-the-Art Comparison: Argoverse 2 (All categories).** We compare our range-view model against different state-of-the-art, peer-reviewed methods on the Argoverse 2 *validation* dataset. This table includes all categories — some which were omitted due to space in the main manuscript.

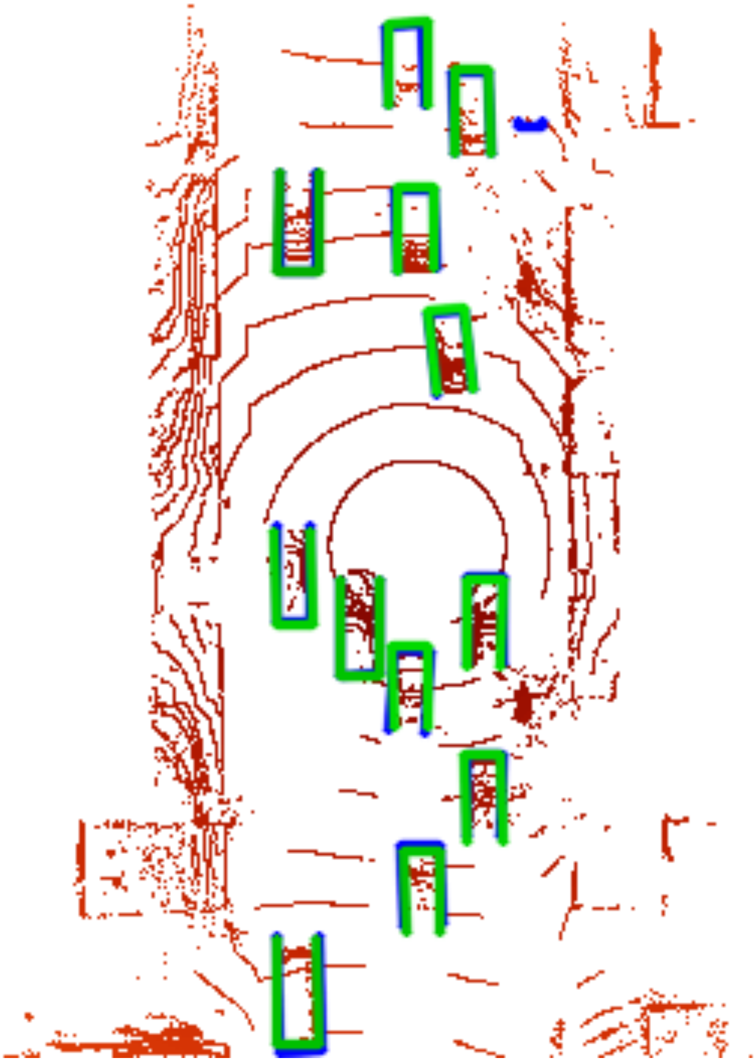

Figure 1: **Qualitative Results: Argoverse 2**. True positives (green) and ground truth cuboids (blue) are shown below for our best performing model. True positives are shown using a $2\,\mathrm{m}$ Euclidean distance from the ground truth cuboid center.

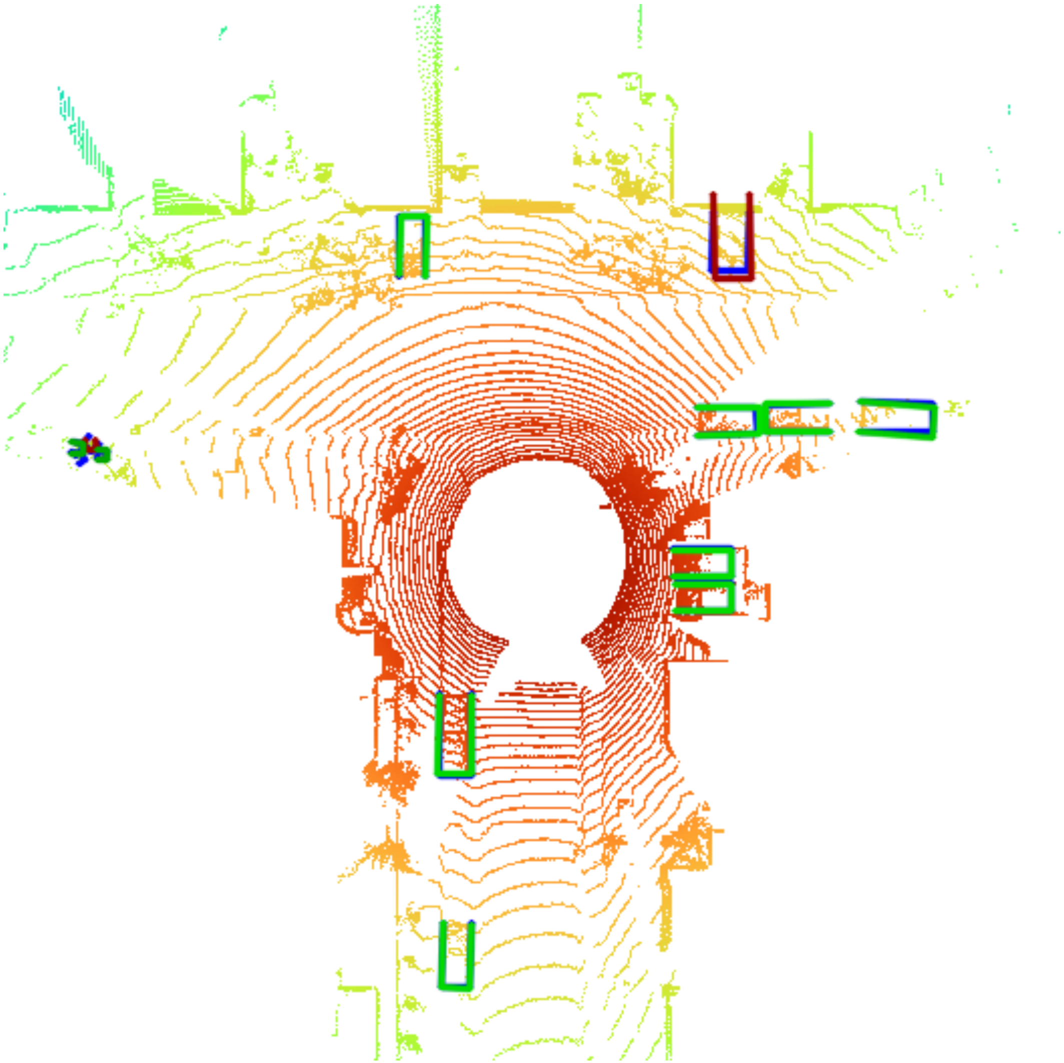

Figure 2: **Qualitative Results: Waymo Open**. True positives (green), false positives (red) and ground truth cuboids (blue) are shown below for our best performing model. True positives are shown using a 0.5 IoU threshold.

References

[1] H. Zhang, Y. Wang, F. Dayoub, and N. Sunderhauf. VarifocalNet: An IoU-Aware Dense Object Detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8514–8523, 2021.

[2] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. Kaesemodel Pontes, D. Ramanan, P. Carr, and J. Hays. Argoverse 2: Next Generation Datasets for Self-Driving Perception and Forecasting. *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 1, Dec. 2021.

[3] L. Fan, X. Xiong, F. Wang, N. Wang, and Z. Zhang. RangeDet: In Defense of Range View for LiDAR-Based 3D Object Detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2918–2927, 2021.

[4] F. Yu, D. Wang, E. Shelhamer, and T. Darrell. Deep Layer Aggregation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2403–2412, 2018.

[5] D. Kalamkar, D. Mudigere, N. Mellempudi, D. Das, K. Banerjee, S. Avancha, D. T. Vooturi, N. Jammalamadaka, J. Huang, H. Yuen, et al. A study of bfloat16 for deep learning training. *arXiv preprint arXiv:1905.12322*, 2019.

[6] Y. Bai, B. Fei, Y. Liu, T. Ma, Y. Hou, B. Shi, and Y. Li. Rangeperception: Taming lidar range view for efficient and accurate 3d object detection. *Advances in Neural Information Processing Systems*, 36, 2024.

[7] T. Yin, X. Zhou, and P. Krahenbuhl. Center-Based 3D Object Detection and Tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11784–11793, 2021.

[8] L. Fan, F. Wang, N. Wang, and Z. Zhang. Fully Sparse 3D Object Detection, Oct. 2022.

[9] Y. Chen, J. Liu, X. Zhang, X. Qi, and J. Jia. Voxelnext: Fully sparse voxelnet for 3d object detection and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21674–21683, 2023.