Harnessing Uncertainty-aware Bounding Boxes for Unsupervised 3D Object Detection

Ruiyang Zhang [1] Hu Zhang[2] Zhedong Zheng[1*]
[1] FST and ICI, University of Macau, China [2] CSIRO Data61, Australia
{yc47931, zhedongzheng}@um.edu.mo, hu1.zhang@csiro.au

Abstract

Unsupervised 3D object detection aims to identify objects of interest from unlabeled raw data, such as LiDAR points. Recent approaches usually adopt pseudo 3D bounding boxes (3D bboxes) from clustering algorithm to initialize the model training. However, pseudo bboxes inevitably contain noise, and such inaccuracies accumulate to the final model, compromising the performance. Therefore, in an attempt to mitigate the negative impact of inaccurate pseudo bboxes, we introduce a new uncertainty-aware framework for unsupervised 3D object detection, dubbed UA3D. In particular, our method consists of two phases: uncertainty estimation and uncertainty regularization. (1) In the uncertainty estimation phase, we incorporate an extra auxiliary detection branch alongside the original primary detector. The prediction disparity between the primary and auxiliary detectors could reflect fine-grained uncertainty at the box coordinate level. (2) Based on the assessed uncertainty, we adaptively adjust the weight of every 3D bbox coordinate via uncertainty regularization, refining the training process on pseudo bboxes. For pseudo bbox coordinate with high uncertainty, we assign a relatively low loss weight. Extensive experiments verify that UA3D is robust against the noisy pseudo bboxes, yielding substantial improvements on nuScenes and Lyft compared to existing approaches, with increases of +3.9% AP_{BEV} *and +1.5%* AP_{3D} *on nuScenes, and +2.3%* AP_{BEV} *and +1.8%* AP_{3D} *on Lyft.*

1. Introduction

Unsupervised 3D object detection [20, 24, 41], given a 3D point cloud, is to identify objects of interest according to the point locations without relying on manual annotations [43, 50, 53, 54], largely saving extra costs and time [25]. The applications span various domains, including autonomous driving [8, 32, 51, 58], traffic management [26, 34], and pedestrian safety [5, 6]. Existing unsupervised 3D object detection works generally follow a self-paced paradigm [54], *i.e.*, estimating some initial pseudo boxes and then iteratively updating both the pseudo label sets and the model weights [50, 52]. However, we observe that the initial pseudo boxes inevitably contain misalignments (see Fig. 1 (a, b)). The accuracy of the pseudo boxes is significantly affected by the inherent characteristics of LiDAR point clouds, such as point sparsity, object proximity, and unclear boundaries between foreground objects and the background. In particular, large and nearby objects are usually easy to detect, and thus most estimated pseudo bboxes are accurate. In contrast, most small, distant objects with less sensor information pose inaccurate pseudo bboxes at the beginning. Without rectifying such erroneous pseudo bboxes, the wrong predictions can be accumulated, consistently compromising self-paced training (see Fig. 1 (c)).

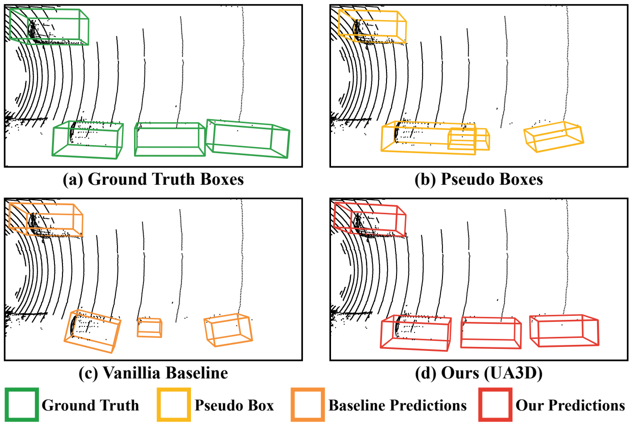

(a) Ground Truth Boxes
(b) Pseudo Boxes
(c) Vanillia Baseline
(d) Ours (UA3D)

Figure 1. **Our Motivation.** Pseudo boxes generated by clustering-based algorithms often contain noise (comparing (a) and (b)). Previous methods [50, 53, 54] directly utilize those noisy pseudo boxes to train detection model, leading to suboptimal performance (see (c)). In contrast, we introduce uncertainty-aware pseudo boxes by assigning coordinate-level uncertainty. High uncertainty is assigned to inaccurate coordinates, and during training, the weights of these uncertain coordinates are adaptively reduced. This approach mitigates the negative impact of noisy pseudo boxes, yielding robust detection (comparing (c) and (d)).

[*] Correspondence to zhedongzheng@um.edu.mo.

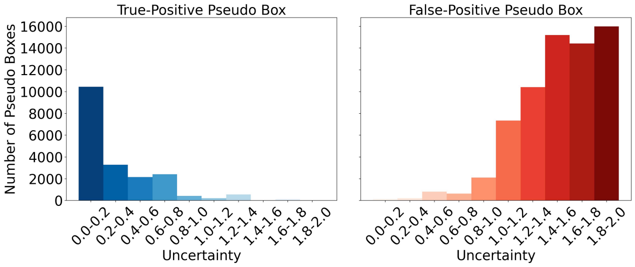


Figure 2. **Statistical Overview of Our Estimated Uncertainty on nuScenes.** Generally, UA3D reliably assigns low uncertainty to accurate pseudo boxes and high uncertainty for noisy ones. For illustration, we average coordinate-level uncertainty to box level.

To mitigate the adverse impacts of inaccurate pseudo bboxes during iterative updates, we introduce **U**ncertainty-**A**ware bounding boxes for unsupervised **3D** object detection (**UA3D**). As the name implies, we explicitly conduct the uncertainty estimation [7, 12, 16] for every pseudo bbox quality (see Fig. 2). The proposed framework consists of two phases: uncertainty estimation and uncertainty regularization. **(1)** In the uncertainty estimation phase, we introduce an auxiliary branch into the existing detection model, attaching to an intermediate layer of the 3D feature extraction backbone. This branch differs from the original primary detection branch in terms of the number of channels. The uncertainty is assessed by comparing the box predictions from primary and auxiliary detectors. Notably, fine-grained uncertainty estimation on coordinate level is achieved by comparing 7 box coordinates of predictions, *i.e*., position (x, y, z coordinates), length, width, height, and rotation, from two detectors. **The intuition is that if the pseudo bboxes are with high uncertainty, two detection branches will lead to prediction discrepancy during training procedure.** We could explicitly leverage such discrepancy as the uncertainty indicator. **(2)** In the uncertainty regularization phase, we adjust the loss weights of different pseudo box coordinates based on the estimated uncertainty during iterative training process. Specifically, with the obtained coordinate-level certainty, the sub-loss computed from each box coordinate is divided by its corresponding uncertainty. Meanwhile, to prevent the model from predicting high uncertainty for all samples, the uncertainty value is also added to the sub-loss for each coordinate. This strategy effectively regularizes the iterative training process from noisy pseudo boxes on coordinate level (see Fig. 1 (d)). For example, if a pseudo box is imprecise in its length but accurate in other coordinates, uncertainty is elevated only for length, thereby reducing loss for that specific coordinate. Quantitative experiments on nuScenes [2] and Lyft [11] validate effectiveness of our method, which consistently outperforms existing approaches. Qualitative analyses reveal that our model generates robust box estimations and achieves higher recall on challenging samples. Furthermore, uncertainty visualization confirms the correlation between high estimated uncertainty and inaccurate pseudo box coordinates. Our contributions are:

- To mitigate negative effects of inaccurate pseudo boxes for unsupervised 3D object detection, we introduce fine-grained uncertainty estimation to assess the quality of pseudo boxes in a learnable manner. Following this, we leverage the estimated uncertainty to regularize the iterative training process, realizing the coordinate-level adjustment in optimization.
- Quantitative experiments on nuScenes [2] and Lyft [11] validate the efficacy of our uncertainty-aware framework, yielding consistent improvements of 3.9% in AP$_{BEV}$ and 1.5% in AP$_{3D}$ on nuScenes, and 2.3% in AP$_{BEV}$ and 1.8% in AP$_{3D}$ on Lyft, compared with existing methods. Qualitative analysis further verifies that our uncertainty estimation successfully identifies inaccuracies in pseudo bounding boxes.

2. Methodology

2.1. Fine-Grained Uncertainty Estimation

Our approach of uncertainty estimation employs an auxiliary detector architecture (see Fig. 3). Typically, 3D object detection models consist of 3D backbone extracting features from point clouds, and 3D detection heads to generate predicted 3D boxes from these features. We introduce an additional 3D detection branch appended to an intermediate layer of the feature extraction backbone. The auxiliary branch mirrors the structure of original branch but differs in channel configuration. We refer to this branch as the auxiliary detector and the original branch is termed the primary detector. We estimate uncertainty as the prediction difference between these two detectors, which can be considered as the degree of disagreement between two different minds. In practice, we use the dense outputs from both detectors, which provide point-wise box predictions across the entire point cloud. For uncertainty estimation, we calculate the ℓ_1 difference between the point-wise predicted boxes of the primary and auxiliary detectors. This difference is computed at the coordinate level to quantify fine-grained uncertainty:

$$\begin{aligned}&\boldsymbol{\Delta_x} = |\boldsymbol{x_p} - \boldsymbol{x_a}|, \boldsymbol{\Delta_y} = |\boldsymbol{y_p} - \boldsymbol{y_a}|, \boldsymbol{\Delta_z} = |\boldsymbol{z_p} - \boldsymbol{z_a}|,\\ &\boldsymbol{\Delta_l} = |\boldsymbol{l_p} - \boldsymbol{l_a}|, \boldsymbol{\Delta_w} = |\boldsymbol{w_p} - \boldsymbol{w_a}|, \boldsymbol{\Delta_h} = |\boldsymbol{h_p} - \boldsymbol{h_a}|,\\ &\boldsymbol{\Delta_\theta} = |\boldsymbol{\theta_p} - \boldsymbol{\theta_a}|,\end{aligned} \tag{1}$$

where $\boldsymbol{x_p}, \boldsymbol{y_p}, \boldsymbol{z_p}, \boldsymbol{l_p}, \boldsymbol{w_p}, \boldsymbol{h_p}, \boldsymbol{\theta_p} \in \mathbb{R}^{n\times 1}$ refer to different coordinate vectors of primary detector dense prediction, namely x, y, z for 3D position, length, width, height, and orientation, $\boldsymbol{x_a}, \boldsymbol{y_a}, \boldsymbol{z_a}, \boldsymbol{l_a}, \boldsymbol{w_a}, \boldsymbol{h_a}, \boldsymbol{\theta_a} \in \mathbb{R}^{n\times 1}$ denote coordinate vectors of auxiliary detector dense prediction, $\boldsymbol{\Delta_x}, \boldsymbol{\Delta_y}, \boldsymbol{\Delta_z}, \boldsymbol{\Delta_l}, \boldsymbol{\Delta_w}, \boldsymbol{\Delta_h}, \boldsymbol{\Delta_\theta} \in \mathbb{R}^{n\times 1}$

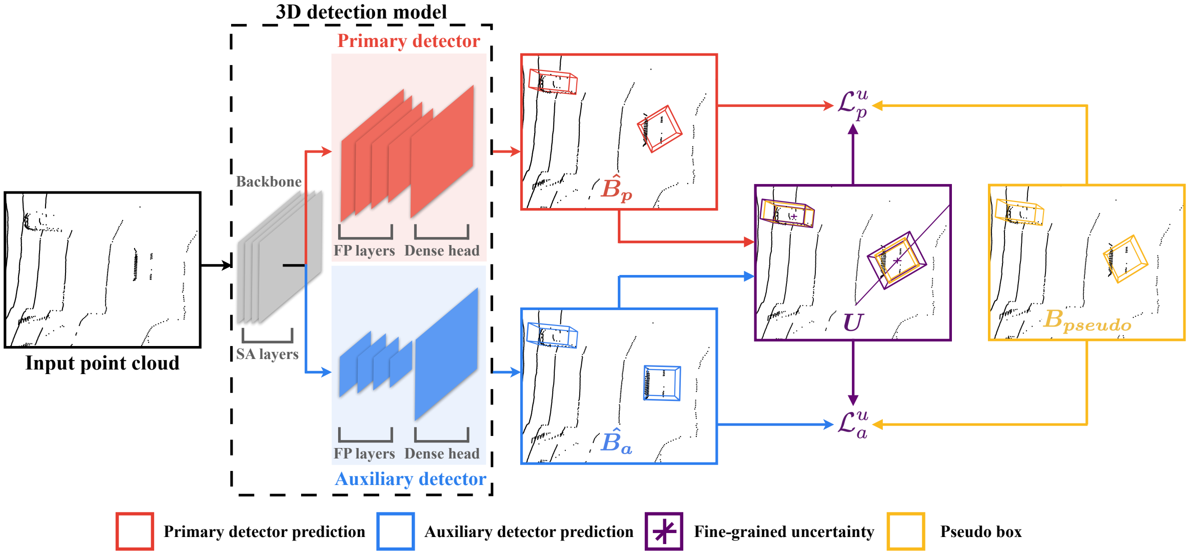


Figure 3. **Overall Pipeline.** Given an input point cloud, an auxiliary detector predicts the bounding boxes $\hat{\boldsymbol{B}}_{\boldsymbol{a}}$ concurrently with the primary detector predictions $\hat{\boldsymbol{B}}_{\boldsymbol{p}}$. We leverage the discrepancy between the two detector predictions as the uncertainty indicator $\boldsymbol{U}$. Specifically, high coordinate-level uncertainty is assigned to inaccurate pseudo box coordinates. For uncertainty regularization, the original detection loss is rectified by the estimated uncertainty as $\mathcal{L}_p^u$ and $\mathcal{L}_a^u$, reducing the weight of inaccurate pseudo boxes on coordinate level. Note: SA refers to Set Abstraction, and FP refers to Feature Propagation. For *uncertainty visualization*, **purple box** represents the uncertainty of length, width, and height, *i.e.*, $\boldsymbol{\Delta_l}$, $\boldsymbol{\Delta_w}$, and $\boldsymbol{\Delta_h}$; **purple orthogonal lines** indicate the uncertainty of the x, y, and z positions, *i.e.*, $\boldsymbol{\Delta_x}$, $\boldsymbol{\Delta_y}$, and $\boldsymbol{\Delta_z}$; and **purple diagonal line** denotes the uncertainty of orientation, *i.e.*, $\boldsymbol{\Delta_\theta}$. In this example, orientation of pseudo box on the right is inaccurate. Our method assigns high uncertainty for the orientation and reduces its weight during model training.

are estimated uncertainty vectors of different coordinates based on prediction discrepancy between two detectors, and n indicates the number of boxes which is same as the number of points in the point cloud. Furthermore, $\hat{\boldsymbol{B}}_{\boldsymbol{p}} = [\boldsymbol{x_p}, \boldsymbol{y_p}, \boldsymbol{z_p}, \boldsymbol{l_p}, \boldsymbol{w_p}, \boldsymbol{h_p}, \boldsymbol{\theta_p}] \in \mathbb{R}^{n\times7}$ refers to primary detector dense predictions, $\hat{\boldsymbol{B}}_{\boldsymbol{a}} = [\boldsymbol{x_a}, \boldsymbol{y_a}, \boldsymbol{z_a}, \boldsymbol{l_a}, \boldsymbol{w_a}, \boldsymbol{h_a}, \boldsymbol{\theta_a}] \in \mathbb{R}^{n\times7}$ denotes auxiliary detector dense predictions, and $\boldsymbol{U} = [\boldsymbol{\Delta_x}, \boldsymbol{\Delta_y}, \boldsymbol{\Delta_z}, \boldsymbol{\Delta_l}, \boldsymbol{\Delta_w}, \boldsymbol{\Delta_h}, \boldsymbol{\Delta_\theta}] \in \mathbb{R}^{n\times7}$ represents the estimated fine-grained uncertainty. Notably, each coordinate of the 3D box is assigned an estimated value, which reflects the uncertainty of that specific coordinate.

Discussions. Why utilize an auxiliary detector to estimate uncertainty, instead of directly regressing uncertainty, as done in previous works [3, 10]? We have studied the additional channel method, which involves introducing extra channels to regress the uncertainty. However, this approach did not yield satisfactory results, as it suffers from overfitting issues, such as predicting zero uncertainty for all samples or uniformly high uncertainty. We attribute this to the inherent complexity of unsupervised 3D detection: simply adding extra channels introduces too few model parameters to effectively capture uncertainty, which is insufficient to manage the complexities involved.

2.2. Adaptive Uncertainty Regularization

Our objective is to adaptively reduce the negative effects of inaccurate pseudo boxes at coordinate level. To achieve this, we rectify original detection loss by incorporating our estimated uncertainty:

$$\mathcal{L}_p^u = \sum_{i=1}^{7}(\frac{\mathcal{L}_{p,i}}{\exp{(\boldsymbol{U_i})}} + \lambda\cdot\boldsymbol{U_i}), \mathcal{L}_a^u = \sum_{i=1}^{7}(\frac{\mathcal{L}_{a,i}}{\exp{(\boldsymbol{U_i})}} + \lambda\cdot\boldsymbol{U_i}), \tag{2}$$

where $\mathcal{L}_p^u, \mathcal{L}_a^u$ denote the uncertainty-regularized loss of primary and auxiliary detectors. For brevity, we represent 7 coordinates of 3D box (see Eq. 1) by $i = 1, 2, ..., 7$. $\mathcal{L}_{p,i}, \mathcal{L}_{a,i}$ represent the original dense head losses of primary and auxiliary detectors for the i-th coordinate, which are calculated by the ℓ_1 loss between corresponding coordinate of the predicted boxes and pseudo boxes. Specifically, $\mathcal{L}_{p,i} = |\hat{\boldsymbol{B}}_{\boldsymbol{p,i}} - \boldsymbol{B_{pseudo,i}}|, \mathcal{L}_{a,i} = |\hat{\boldsymbol{B}}_{\boldsymbol{a,i}} - \boldsymbol{B_{pseudo,i}}|$, where $\boldsymbol{B_{pseudo,i}} \in \mathbb{R}^{n\times1}$ is the i-th coordinate of assigned dense pseudo boxes. $\boldsymbol{U_i}$ denotes the estimated fine-grained uncertainty of the corresponding coordinate in $\boldsymbol{U}$. To prevent divide-by-zero errors and stabilize the learning process, we normalize estimated uncertainty with exponential function. Additionally, we incorporate term $\lambda\cdot\boldsymbol{U_i}$ to prevent the model from consisting predicting high uncertainty, where λ controls penalty strength. Empirically, when uncertainty of certain coordinate is high, weight of that inaccurate pseudo box coordinate is diminished, thereby reducing its impact on training process. Conversely, when uncertainty is low, for instance, nearing zero, the loss reverts to original detection loss, preserving the full influence of that pseudo box coordinate. As a result, our uncertainty regularization dynamically mitigates negative effects of inaccurate pseudo boxes on coordinate level.

The regularization process is uniformly applied to both primary and auxiliary detectors. Each detector takes into account the prediction of the other and adjusts weights of pseudo box coordinates accordingly, who diminishes influence of pseudo box coordinates when significant prediction disagreement is evident, and reserves impact of pseudo box coordinates when two predictions concur. Therefore, the final loss $\mathcal{L}_{total}$ can be formulated as:

$$\mathcal{L}_{total} = \mathcal{L}_p^u + \mu \cdot \mathcal{L}_a^u, \qquad (3)$$

where $\mathcal{L}_p^u$ is the uncertainty-regularized loss for the primary detector, $\mathcal{L}_a^u$ is the uncertainty-regularized loss for the auxiliary detector, μ denotes the auxiliary detector loss weight.

Discussions. Why not leverage rule-based methods to pre-calculate box uncertainty? Our uncertainty-aware framework is learnable and more adaptive (see Tab. 3). Uncertainty can be caculated by human-observed knowledge, *e.g.*, the further the box, the higher the uncertainty. However, such rules can lead to errors. For example, a distant box can be very accurate, but under rule-based uncertainty, its influence can be unjustly diminished, potentially degrading model performance. Our learnable uncertainty avoids this pitfall by not only assimilating human-observed rules and knowledge but also adaptively handling different cases. For instance, if a distant pseudo box is very accurate, both the primary and auxiliary detectors can provide similar predictions, resulting in low uncertainty and ensuring that the box is appropriately valued during training.

3. Experiment

3.1. Settings

Datasets. Our experiments are conducted on nuScenes [2] and Lyft [11]. nuScenes consists of 1,000 complex scenes collected in urban environments. Each scene sequence contains 20 seconds data. In total, nuScenes contains 400K key sample and 390K LiDAR point clouds. Lyft includes 1,000 hours driving data, 170,000 scenes (25 seconds per scene), covering both LiDAR point cloud and image data. It is worth noting that we do not use any ground truth 3D boxes during the training phase and ground truth boxes are exclusively used for evaluation.

Backbone. PointRCNN utilizes PointNet++ [31] for extracting point-wise features from the LiDAR point clouds. Within PointNet++, Set Abstraction layers first perform point grouping and local feature extraction, Feature Propagation layers then conduct feature upsampling and propagate abstract features back to point-wise representation. Following this, dense head predicts a 3D box for each point based on these extracted features. Lastly, region of interest (ROI) head aggregates object proposals from the point-wise predictions into final predictions.

Implementation Details. Channel numbers in the original Feature Propagation layers are (C_1, C_2, C_3, C_4), while in the auxiliary Feature Propagation layers, they are scaled to ($\gamma \cdot C_1$, $\gamma \cdot C_2$, $\gamma \cdot C_3$, $\gamma \cdot C_4$), where γ represents ratio coefficient. We then integrate a new dense head and ROI head after the introduced Feature Propagation layers to establish the auxiliary detector. For both nuScenes and Lyft, the uncertainty regularization coefficient λ is set to $1e^{-5}$. For the hyper-parameters, we only tune them on nuScenes and directly apply them to Lyft. During training, we follow the self training paradigm established by previous work MODEST [50]. Specifically, we conduct seed training and 10 rounds of self training in all our experiments.

3.2. Comparison with State-of-the-Art Methods

Quantitative Results on nuScenes. We present the results for nuScenes [2] in Table 1. UA3D outperforms the state-of-the-art method LiSe [54] by 3.9% in AP$_{BEV}$ and 1.5% in AP$_{3D}$ under LiDAD-based setting. Those general performance enhancement underscores the efficacy of our proposed fine-grained uncertainty estimation and adaptive uncertainty regularization in refining learning process from noisy pseudo boxes. It confirms that reducing the negative impact of inaccurate pseudo boxes on coordinate level can significantly boost model detection performance. Notably, for objects in the long-range (50-80m), AP$_{BEV}$ sees a remarkable increase. This significant boost is attributed to the typically lower accuracy of long-range pseudo boxes, where uncertainty plays a pivotal role in dynamically adjusting the weights of pseudo boxes coordinates according to their varying qualities. Moreover, we observe consistent improvement in LiDAR-image fusion settings. The clear improvement on both LiDAR-based MODEST and LiDAR-image-based method LiSe shows our method is compatible with various baseline methods.

Quantitative Results on Lyft. We further conduct experiments on Lyft [11] (see Table 2). Our uncertainty-aware method surpasses baselines by 2.3% in AP$_{BEV}$ and 1.8% in AP$_{3D}$ under LiDAR-based setting, and 2.5% in AP$_{BEV}$ and 1.7% in AP$_{3D}$ under LiDAR-based setting. Notably, we use the same hyper-parameter settings as those in nuScenes experiments, validating the generalizability and effectiveness of our uncertainty-aware approach. We observe the most prominent improvements are from the long-range (50m-80m), which verifies the efficacy of our method in enhancing the detection capability of distant objects. These objects are typically challenging to recognize.

3.3. Ablation Studies and Further Discussion

Comparison with Other Uncertainty Mechanism. We compare our proposed learnable uncertainty-aware method with rule-based, regression-based, ensemble-based, and Monte Carlo Dropout-based uncertainty to validate the superiority of our learnable approach (see Table 3). The rule-based baselines are motivated by CPD [44] and [17]. The

Method	Conference	Data	Round	0m-30m		30m-50m		50m-80m		0m-80m	
				AP_{BEV}	AP_{3D}	AP_{BEV}	AP_{3D}	AP_{BEV}	AP_{3D}	AP_{BEV}	AP_{3D}
Supervised [50]	-	-	-	39.8	34.5	12.9	10.0	4.4	2.9	22.2	18.2
LiDAR-Based											
MODEST [50]	CVPR'22	L	0	16.5	12.5	1.3	0.8	0.3	0.1	7.0	5.0
MODEST [50]	CVPR'22	L	10	24.8	17.1	5.5	1.4	1.5	0.5	11.8	6.6
OYSTER [53]	CVPR'23	L	0	14.7	12.3	1.5	1.1	0.5	0.3	6.2	5.4
OYSTER [53]	CVPR'23	L	2	26.6	21.3	4.4	1.8	1.7	0.4	12.7	8.0
LiSe [54]	ECCV'24	L	0	14.8	12.3	1.5	0.4	0.4	0.2	6.1	4.2
LiSe [54]	ECCV'24	L	10	31.4	21.1	7.0	2.5	2.6	0.5	15.7	9.0
UA3D (ours)	-	L	0	13.7	11.5	0.9	0.6	0.5	0.2	5.4	4.9
UA3D (ours)	-	L	2	30.1	19.8	7.8	2.9	3.1	0.5	15.1	9.1
UA3D (ours)	-	L	10	**38.3**	**23.8**	**10.1**	**3.5**	**4.3**	**0.7**	**19.6**	**10.5**
LiDAR-Image Fusion											
LiSe [54]	ECCV'24	L & I	0	5.8	4.7	0.6	0.2	0.3	0.2	2.1	1.8
LiSe [54]	ECCV'24	L & I	10	35.0	24.0	11.4	4.4	4.8	1.3	19.8	11.4
UA3D (ours)	-	L & I	0	8.4	7.3	0.8	0.5	0.4	0.8	3.5	2.4
UA3D (ours)	-	L & I	10	**38.2**	**24.7**	**12.5**	**4.9**	**5.0**	**1.7**	**21.3**	**12.1**

Table 1. **Quantitative Results on nuScenes [2].** UA3D significantly surpasses the state-of-the-art LiSe [54] across all evaluated metrics. This validates the efficacy of proposed coordinate-level uncertainty estimation and regularization in mitigating negative impacts of noisy pseudo boxes, thereby enhancing detection performance. We report AP_{BEV} / AP_{3D} at IoU=0.25. 'L' for LiDAR data and 'I' for image data. Round refers to the number of self-training round. The best results are in **bold**, and the second-best results are underlined.

Method	Conference	Data	Round	0m-30m		30m-50m		50m-80m		0m-80m	
				AP_{BEV}	AP_{3D}	AP_{BEV}	AP_{3D}	AP_{BEV}	AP_{3D}	AP_{BEV}	AP_{3D}
Supervised [50]	-	-	-	82.8	82.6	70.8	70.3	50.2	49.6	69.5	69.1
LiDAR-Based											
MODEST-PP [50]	CVPR'22	L	0	46.4	45.4	16.5	10.8	0.9	0.4	21.8	18.0
MODEST-PP [50]	CVPR'22	L	10	49.9	49.3	32.3	27.0	3.5	1.4	30.9	27.3
MODEST [50]	CVPR'22	L	0	65.7	63.0	41.4	36.0	8.9	5.7	42.5	37.9
MODEST [50]	CVPR'22	L	10	73.8	71.3	62.8	60.3	27.0	24.8	57.3	55.1
LiSe [54]	ECCV'24	L	0	42.9	42.6	11.0	10.7	0.5	0.4	20.0	19.6
LiSe [54]	ECCV'24	L	10	76.0	73.4	64.7	61.8	28.5	24.9	59.8	56.1
UA3D (ours)	-	L	0	66.0	63.3	43.8	36.3	8.9	5.1	43.2	38.0
UA3D (ours)	-	L	10	**76.5**	**73.6**	**64.6**	**62.0**	**36.8**	**29.0**	**62.1**	**57.9**
LiDAR-Image Fusion											
LiSe [54]	ECCV'24	L & I	0	54.5	54.0	24.2	22.8	1.4	1.2	29.2	27.5
LiSe [54]	ECCV'24	L & I	10	76.7	74.0	66.1	64.4	46.6	43.7	65.6	62.5
UA3D (ours)	-	L & I	0	60.3	57.4	35.5	28.6	2.4	2.5	35.8	31.1
UA3D (ours)	-	L & I	10	**78.2**	**74.6**	**67.3**	**65.1**	**49.2**	**46.0**	**68.1**	**64.2**

Table 2. **Quantitative Results on Lyft [11].** UA3D outperforms LiSe [54] by a clear margin, under both LiDAR-based and LiDAR-image fusion settings. Notably, we employ same hyper-parameters as those in nuScenes, validating robustness of UA3D across different datasets.

former one estimates the quality score of pseudo boxes based on distance and point number within box grids. The latter one utilizes the volume ratio to estimate box confidence. For distance-rule uncertainty, the uncertainty of a pseudo box is quantified as $u = \frac{\min(b_x, \tau_x)}{\tau_x}$, where b_x denotes the distance of the box from the ego vehicle, and τ_d represents the selected distance threshold. We empirically set $\tau_x = 100m$. For Num. Point-rule uncertainty, the uncertainty is formulated as $u = \frac{\tau_n}{\min(b_{num\_pts}, \tau_n)}$, where b_{num_pts} refers to the number of points within the 3D pseudo box, and τ_n is the selected points threshold set at $\tau_n = 100$. For Volume-rule uncertainty, the uncertainty is computed as $u = \frac{\tau_v}{\min(b_l \cdot b_w \cdot b_h, \tau_v)}$, where b_l, b_w, and b_h indicate the length, width, and height of the 3D pseudo box, and τ_v is the chosen volume threshold set at $\tau_v = 10m^3$. UA3D outperforms all rule-based uncertainties by effectively addressing scenarios where rule-based approaches fail. Our learnable uncertainty is capable of assigning high uncertainty to challenging cases due to prediction discrepancies between the primary and auxiliary detectors. We also conduct comparisons with additional uncertainty estimation methods, *e.g.*, regression-based uncertainty, ensemble methods, and MC Dropout methods. Our prediction discrepancy-based method consistently outperforms all these baselines. Specifically, we find that regression-based uncertainty suffers from overfitting, often predicting either all zeros or uniformly high uncertainty. We attribute this to the complexity of unsupervised 3D detection, where adding extra channels introduces too few model parameters to effectively capture uncertainty. For ensemble and MC Dropout methods, their performance is limited by the pretrained detection model. Moreover, they typically require around 10–20 ensemble members or 10–20 inference passes, resulting in significantly higher memory and computation costs. In contrast, our prediction discrepancy approach only requires a single forward pass to obtain the final uncertainty, making it much more efficient and effective.

Ablation of Uncertainty Granularity. We present an abla-

Method	0m-30m		30m-50m		50m-80m		0m-80m	
	BEV	3D	BEV	3D	BEV	3D	BEV	3D
Rule-Based								
Distance Rule	29.6	19.6	7.2	2.2	3.2	0.5	14.8	8.1
Volume Rule	25.7	17.7	5.6	2.2	2.5	0.4	12.3	7.4
Num. Point Rule	27.3	17.6	7.3	2.8	2.3	0.3	13.7	7.5
Regression-Based								
Additional Channel	26.3	18.8	4.9	2.2	2.0	0.3	12.1	7.7
Additional FC	27.2	19.7	4.0	1.9	1.2	0.1	12.5	8.1
Ensemble-Based								
10 Members	32.5	20.7	5.5	2.3	3.1	0.4	15.0	8.6
20 Members	32.1	23.8	10.1	3.5	3.6	0.7	15.3	9.1
Monte Carlo Dropout-Based								
$p = 0.1, N = 10$	29.6	19.6	7.2	2.2	3.2	0.2	14.8	8.1
$p = 0.2, N = 20$	28.1	20.3	8.0	3.3	3.9	0.5	15.7	9.2
UA3D (ours)	**38.3**	**23.8**	**10.1**	**3.5**	**4.3**	**0.7**	**19.6**	**10.5**

Table 3. **Comparison with Other Uncertainty.** Our learnable uncertainty surpasses all other types of uncertainty, validating its superiority in handling complex cases. Results are from nuScenes. BEV is short for AP$_{BEV}$, and 3D for AP$_{3D}$.

Granularity	0m-30m		30m-50m		50m-80m		0m-80m	
	BEV	3D	BEV	3D	BEV	3D	BEV	3D
Point cloud-level	27.7	18.7	3.6	1.2	1.2	0.1	12.1	6.7
Box-level	34.9	24.6	7.5	2.8	3.6	0.1	17.2	9.9
Coordinate-level	**38.3**	**23.8**	**10.1**	**3.5**	**4.3**	**0.7**	**19.6**	**10.5**

Table 4. **Ablation of Uncertainty Granularity.** We find that our proposed coordinate-level uncertainty outperforms other coarse-grained uncertainty, such as box-level and point cloud-level. By addressing inaccurate box coordinates individually, we mitigate the negative impact of noisy pseudo boxes adaptively. Our default setting is marked with gray .

γ	0m-30m		30m-50m		50m-80m		0m-80m	
	BEV	3D	BEV	3D	BEV	3D	BEV	3D
0.25	32.6	23.5	8.6	3.1	4.3	0.2	16.9	9.9
0.5	**38.3**	**23.8**	**10.1**	**3.5**	4.3	**0.7**	**19.6**	**10.5**
1	29.6	22.3	6.0	2.3	3.3	0.1	14.7	8.5
2	29.5	20.5	7.9	3.0	**4.4**	0.3	15.8	8.9

Table 5. **Ablation of Channel Number Ratio between Auxiliary and Primary Detector.** γ denotes the channel number coefficient of the auxiliary detector, with the best performance achieved at 0.5. Default setting is in gray .

λ	0m-30m		30m-50m		50m-80m		0m-80m	
	BEV	3D	BEV	3D	BEV	3D	BEV	3D
$1e^{-4}$	33.8	20.4	6.1	1.5	2.9	0.3	15.2	7.4
$1e^{-5}$	**38.3**	**23.8**	**10.1**	**3.5**	**4.3**	**0.7**	**19.6**	**10.5**
$1e^{-6}$	18.1	13.7	3.2	1.3	1.6	0.2	8.4	5.6

Table 6. **Ablation of Uncertainty Regularization Coefficient λ.** We obtain the best result at $\lambda = 1e^{-5}$, as it ensures uncertainty estimation and regularization play a proper role, preventing the uncertainty from vanishing or exploding. Default setting in gray .

μ	0m-30m		30m-50m		50m-80m		0m-80m	
	BEV	3D	BEV	3D	BEV	3D	BEV	3D
0.25	33.9	22.2	5.5	2.2	2.1	0.3	15.4	8.8
0.5	32.5	20.7	5.5	2.3	3.1	0.4	15.0	8.6
1	**38.3**	**23.8**	**10.1**	**3.5**	**4.3**	**0.7**	**19.6**	**10.5**
2	33.2	20.8	4.9	1.9	2.1	0.3	14.5	8.4

Table 7. **Ablation of Auxiliary Detector Loss Weight μ.** The balanced learning process, *i.e*., equal weights for both detectors, leads to optimal results. Our default setting is in gray .

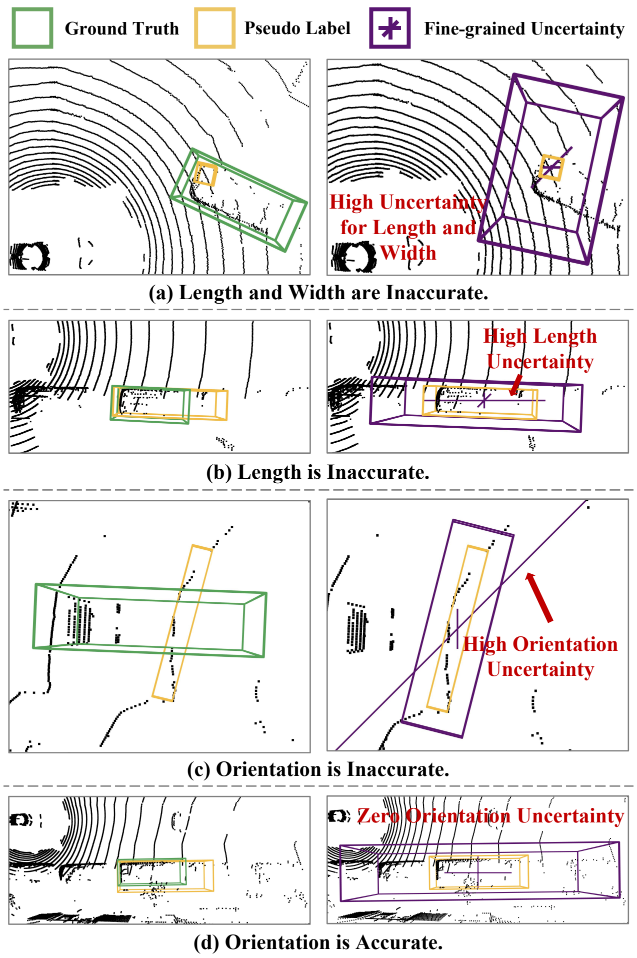
Layer	0m-30m		30m-50m		50m-80m		0m-80m	
	BEV	3D	BEV	3D	BEV	3D	BEV	3D
SA Layer 4	**38.3**	**23.8**	**10.1**	**3.5**	4.3	**0.7**	**19.6**	**10.5**
FP Layer 1	34.4	21.2	9.4	3.1	**4.6**	0.6	18.0	9.3
FP Layer 2	31.3	19.4	6.6	2.1	2.5	0.3	15.1	8.0

Table 8. **Ablation of Backbone Layer Auxiliary Detector Attaches.** From shallow to deeper, we study through SA Layer 4, FP Layer 1, and FP Layer 2. We observe that attaching the auxiliary detector to a shallower layer, *e.g*., the SA Layer 4, yields the best performance. gray line is our default setting.

tion study on the uncertainty granularity in Table 4. For our proposed coordinate-level uncertainty, the uncertainty estimation and regularization is applied at the coordinate level, where the loss weight for each coordinate of each box is adjusted adaptively based on its uncertainty value. For box-level uncertainty, we sum and average the uncertainty values of the 7 coordinates for each box. For point cloud-level uncertainty, we aggregate the uncertainty of all boxes in the point cloud to represent overall uncertainty. We observe that the best results are achieved with our coordinate-level uncertainty. This approach corrects inaccurate pseudo boxes in a more fine-grained and adaptive manner, effectively mitigating the negative impact of noise. In contrast, box-level uncertainty regularization treats the entire box as either certain or uncertain, ignoring differences among the coordinates. The coarse-grained box-level approach can compromise the efficacy of regularization. At the point cloud level, the regularization effect is weak, resulting in performance degradation to the baseline (MODEST).

Design of Uncertainty Estimation. We present an ablation study on the design of the auxiliary detector in Table 5. The configuration with $\gamma = 0.5$ yields the best results. This configuration provides enough model capacity to fit accurate pseudo boxes while avoiding over-fitting to noisy pseudo boxes. As a result, the primary and auxiliary detector predictions tend to diverge for inaccurate pseudo boxes, leading to more effective uncertainty estimation and regularization. $\gamma = 0.25$ indicates a smaller auxiliary detector with weaker capacity in fitting even accurate pseudo boxes. Conversely, larger auxiliary detectors, such as those with $\gamma = 1$ and $\gamma = 2$, exhibit learning capacities similar to primary detector, which diminishes efficacy of uncertainty learning.

Design of Uncertainty Regularization. We explore the ef-


Ground Truth
Pseudo Label
Fine-grained Uncertainty
High Uncertainty for Length and Width
High Length Uncertainty
High Orientation Uncertainty
Zero Orientation Uncertainty

(a) Length and Width are Inaccurate.

(b) Length is Inaccurate.

(c) Orientation is Inaccurate.

(d) Orientation is Accurate.

Figure 4. **Calibration Between Estimated Uncertainty and Errors in Pseudo Boxes.** Our estimated coordinate-level uncertainty closely calibrates with inaccuracies in pseudo box coordinates, with high uncertainty assigned to erroneous pseudo box coordinates. **(a)** Length and width of the pseudo box are inaccurate, the uncertainty for these coordinates is correspondingly high. **(b)** A similar pattern is observed for inaccuracies in the length and x-position coordinates. **(c)** Pseudo box orientation is highly inaccurate, a distinctly high uncertainty is assigned to the orientation. **(d)** Conversely, pseudo box orientation is highly accurate, its associated uncertainty is nearly zero.

fects of varying the uncertainty regularization coefficient λ (see Eq. 2) in Table 6. The optimal performance is observed with $\lambda = 1e^{-5}$, which allows uncertainty estimation and regularization to play a proper role and avoids uncertainty vanishing or explosion. Other settings yield sub-optimal results compared with $\lambda = 1e^{-5}$. A high $\lambda = 1e^{-4}$ imposes a strong penalty for high uncertainty and suppresses the role of uncertainty during training. Conversely, a low $\lambda = 1e^{-6}$, which imposes a minimal penalty for high uncertainty, leads to excessively high uncertainty values across all samples.

Ablation of Auxiliary Detector Loss Weight. We conduct an ablation study on the loss weight μ of auxiliary detector (see Table 7). We observe that $\mu = 1$ yields the best detection performance. This suggests that applying equal weights to both branches fosters a balanced learning process, enhancing overall model performance.

Ablation of Auxiliary Detector Attached Layer. Additionally, we present an ablation study on the feature extraction backbone layer to which the auxiliary detector is attached (see Table 8). When attaching to the sa_layer_4, we utilize all the FP layers, which facilitates the construction of an independent auxiliary detection branch endowed with full capacity. This maximizes the effectiveness of our proposed uncertainty-aware framework.

3.4. Qualitative Analysis

Calibration Between Estimated Uncertainty and Pseudo Label Inaccuracy. We provide visualizations of the calibration between uncertainty and pseudo box inaccuracy in Fig. 4. We observe that our UA3D effectively learns uncertainty that aligns closely with pseudo box inaccuracies at the coordinate level. This alignment facilitates subsequent uncertainty regularization, where pseudo box coordinates with high uncertainty are assigned lower weights.

Qualitative Comparison. We compare the predictions from our uncertainty-aware method against those from MODEST [50] and OYSTER [53] (see Fig. 5). Notably, our method achieves more accurate predictions in terms of shape, location, and orientation (see (a)). Furthermore, we observe an increase in the recall rate, especially for distant and smaller objects (see (b)). The pseudo boxes for these objects are often less reliable due to the challenges in estimating such boxes. Our approach selectively discounts these unreliable boxes, allowing high-quality boxes to play a more prominent role. Consequently, UA3D enhances recall performance for these categories.

4. Related Work

Unsupervised 3D Object Detection. One trajectory focuses on object discovery from LiDAR point clouds [15, 43]. MODEST [50] pioneers the use of multi-traversal method to generate pseudo boxes for moving objects. OYSTER [53] builds on this approach by advocating for learning in a near-to-far fashion. Recently, CPD [43] enhances this methodology by employing precise prototypes for various object classes to boost detection accuracy. The second trajectory focuses on harnessing knowledge from 2D space [48, 54]. Yao *et al.* [48] propose the alignment of concept features from 3D point clouds with semantic data from 2D images. LiSe [54] fuses LiDAR and 2D knowledge to discover the far and small objects. However, owning to the inherent noise in pseudo boxes, the final efficacy of these approaches can be compromised [15, 18, 43, 49]. Different from existing works, we utilize fine-grained uncertainty estimation and regularization to mitigate the negative effect of

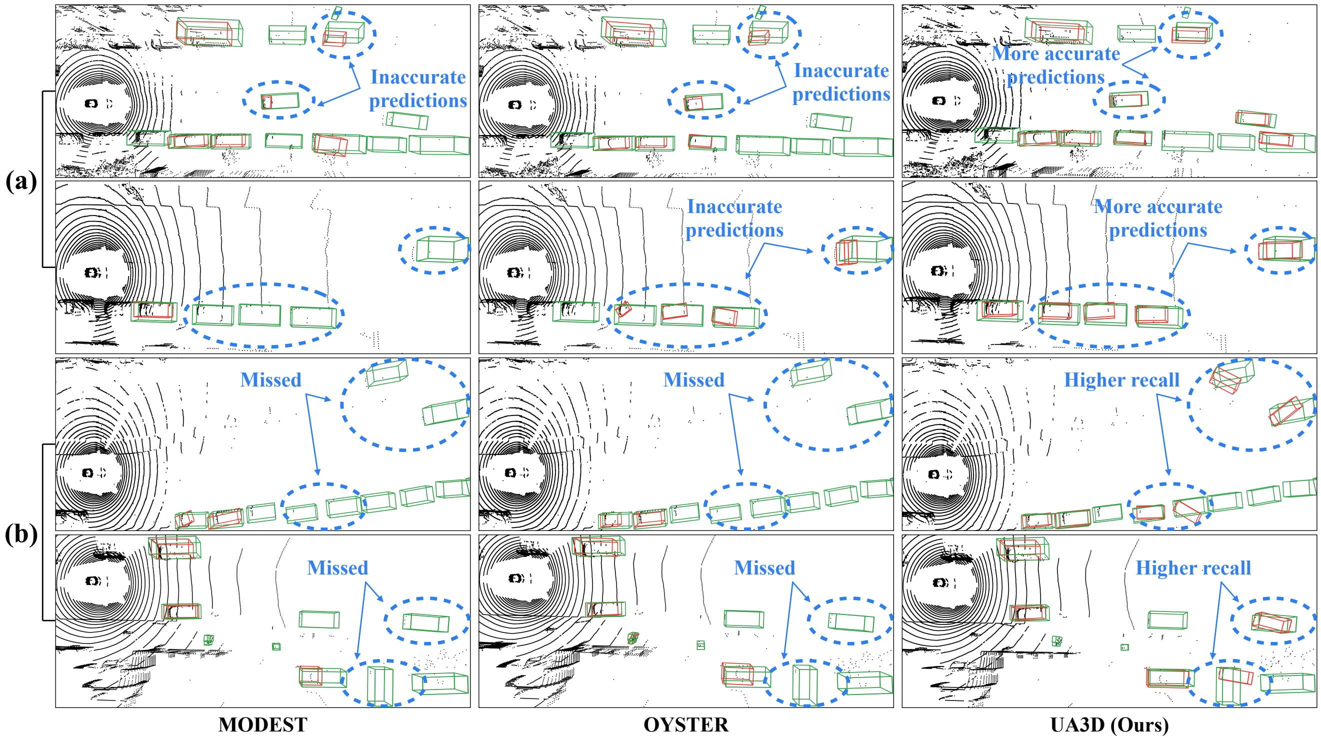


Figure 5. **Qualitative Comparison with Previous Methods.** We compare UA3D with MODEST [50], and OYSTER [53]. (a) Generally, our method shows a clear improvement in box accuracy over previous methods. (b) For some challenging objects with few points or far away, our method can still retain a higher recall rate. **Green** boxes denote ground truth and **red** boxes are predictions.

inaccurate pseudo boxes.

Uncertainty Learning. Single deterministic methods [14, 28, 33, 36] adapt the original model to directly estimate prediction uncertainty, though the extra uncertainty estimation usually compromises the original task. Bayesian methods [21, 27, 29, 42] utilize probabilistic neural networks to estimate uncertainty by assessing the variance across multiple forward passes of the same input, which are limited by high computational costs. Similarly, recent works [55, 56] tune temperatures of large models for uncertainty estimation via multiple inference. Ensemble methods [13, 23, 30, 35, 59] estimate uncertainty through the combined outputs of various deterministic models during inference, aiming primarily to enhance prediction accuracy. Test-time augmentation methods [4, 19, 22, 37] create multiple predictions by augmenting input samples during testing, with primary challenge in selection of appropriate augmentation. Different from existing techniques [7, 9, 57], we devise auxiliary detection branch alongside primary detector to enable quantification of fine-grained uncertainty.

3D Object Detection Framework. Works in this domain can primarily be divided into 3 categories based on point representation [1, 32]. First, voxel-based methods [45, 60] transform unordered point clouds into compact 2D or 3D grids, subsequently compressing them into a bird's-eye view (BEV) 2D representation. These approaches are generally more computationally efficient and hardware-friendly but sacrifice fine-grained details. Second, point-based approaches utilize permutation-invariant operations to directly process the original geometry of raw point clouds [38, 40, 47], thereby excelling in capturing detailed features at the expense of increased model latency. Lastly, voxel-point based methods [39, 46] aim to merge the computational advantages of voxel-based techniques with the detailed accuracy of point-based methods, marking a progressive trend in this field. Notably, UA3D enhances performance of those detection models in unsupervised setting, with fine-grained uncertainty learning.

5. Conclusion

We propose an uncertainty-aware framework that identifies inaccuracies in pseudo boxes at a fine-grained coordinate level and mitigates their negative effects. In the uncertainty estimation phase, we introduce an auxiliary detector to capture the prediction discrepancy between the auxiliary and primary detectors, harnessing these discrepancies as fine-grained indicators of uncertainty. In the uncertainty regularization phase, the estimated uncertainty is utilized to refine the training process, adaptively minimizing the negative impact of inaccurate pseudo boxes. Experiments on nuScenes and Lyft validate our approach, with qualitative results linking high uncertainty to label inaccuracy.

Acknowledgments. We acknowledge supports from the Guangdong Basic and Applied Basic Research Foundation 2025A1515012281, the Nanjing Municipal Science and Technology Bureau 202401035 and the University of Macau MYRG-GRG2024-00077, and Conference Grant – FST, CG2025-FST.

References

[1] Eduardo Arnold, Omar Y Al-Jarrah, Mehrdad Dianati, Saber Fallah, David Oxtoby, and Alex Mouzakitis. A survey on 3d object detection methods for autonomous driving applications. *IEEE Transactions on Intelligent Transportation Systems*, 20(10):3782–3795, 2019. 8

[2] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, pages 11621–11631, 2020. 2, 4, 5, 3

[3] Jiwoong Choi, Dayoung Chun, Hyun Kim, and Hyuk-Jae Lee. Gaussian yolov3: An accurate and fast object detector using localization uncertainty for autonomous driving. In *ICCV*, pages 502–511, 2019. 3

[4] Pedro Conde, Tiago Barros, Rui L Lopes, Cristiano Premebida, and Urbano J Nunes. Approaching test time augmentation in the context of uncertainty calibration for deep neural networks. *arXiv preprint arXiv:2304.05104*, 2023. 8

[5] Tarak Gandhi and Mohan Manubhai Trivedi. Pedestrian protection systems: Issues, survey, and challenges. *IEEE Transactions on intelligent Transportation systems*, 8(3):413–430, 2007. 1

[6] Dariu M Gavrila, Jan Giebel, and Stefan Munder. Vision-based pedestrian detection: The protector system. In *IEEE Intelligent Vehicles Symposium, 2004*, pages 13–18. IEEE, 2004. 1

[7] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. A survey of uncertainty in deep neural networks. *Artificial Intelligence Review*, 56(Suppl 1):1513–1589, 2023. 2, 8

[8] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of field robotics*, 37(3):362–386, 2020. 1

[9] Wenchong He, Zhe Jiang, Tingsong Xiao, Zelin Xu, and Yukun Li. A survey on uncertainty quantification methods for deep learning, 2024. 8

[10] Yihui He, Chenchen Zhu, Jianren Wang, Marios Savvides, and Xiangyu Zhang. Bounding box regression with uncertainty for accurate object detection. In *CVPR*, pages 2888–2897, 2019. 3

[11] John Houston, Guido Zuidhof, Luca Bergamini, Yawei Ye, Long Chen, Ashesh Jain, Sammy Omari, Vladimir Iglovikov, and Peter Ondruska. One thousand and one hours: Self-driving motion prediction dataset. In *CoRL*, pages 409–418. PMLR, 2021. 2, 4, 5, 3

[12] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *NeurIPS*, 30, 2017. 2

[13] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *NeurIPS*, 30, 2017. 8

[14] Jinsol Lee and Ghassan AlRegib. Gradients as a measure of uncertainty in neural networks. In *2020 ICIP*, pages 2416–2420. IEEE, 2020. 8

[15] Ted Lentsch, Holger Caesar, and Dariu M Gavrila. Union: Unsupervised 3d object detection using object appearance-based pseudo-classes. *NeurIPS*, 2024. 7

[16] Yiping Li, Jianwen Chen, and Ling Feng. Dealing with uncertainty: A survey of theories and practices. *IEEE Transactions on Knowledge and Data Engineering*, 25(11):2463–2482, 2012. 2

[17] YuXuan Liu, Nikhil Mishra, Maximilian Sieb, Yide Shentu, Pieter Abbeel, and Xi Chen. Autoregressive uncertainty modeling for 3d bounding box prediction. In *European Conference on Computer Vision*, pages 673–694. Springer, 2022. 4

[18] Katie Luo, Zhenzhen Liu, Xiangyu Chen, Yurong You, Sagie Benaim, Cheng Perng Phoo, Mark Campbell, Wen Sun, Bharath Hariharan, and Kilian Q Weinberger. Reward fine-tuning for faster and more accurate unsupervised object discovery. *NeurIPS*, 36:13250–13266, 2023. 7

[19] Alexander Lyzhov, Yuliya Molchanova, Arsenii Ashukha, Dmitry Molchanov, and Dmitry Vetrov. Greedy policy search: A simple baseline for learnable test-time augmentation. In *Conference on uncertainty in artificial intelligence*, pages 1308–1317. PMLR, 2020. 8

[20] Xinzhu Ma, Wanli Ouyang, Andrea Simonelli, and Elisa Ricci. 3d object detection from images for autonomous driving: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 1

[21] Yi-An Ma, Tianqi Chen, and Emily Fox. A complete recipe for stochastic gradient mcmc. *NeurIPS*, 28, 2015. 8

[22] Rui Magalhães and Alexandre Bernardino. Quantifying object detection uncertainty in autonomous driving with test-time augmentation. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–7. IEEE, 2023. 8

[23] Andrey Malinin, Bruno Mlodozeniec, and Mark Gales. Ensemble distribution distillation. *arXiv preprint arXiv:1905.00076*, 2019. 8

[24] Jiageng Mao, Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. 3d object detection for autonomous driving: A comprehensive survey. *International Journal of Computer Vision*, 131(8):1909–1963, 2023. 1

[25] Qinghao Meng, Wenguan Wang, Tianfei Zhou, Jianbing Shen, Yunde Jia, and Luc Van Gool. Towards a weakly supervised framework for 3d point cloud object detection and annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4454–4468, 2021. 1

[26] Vicente Milanes, Jorge Villagra, Jorge Godoy, Javier Simó, Joshué Pérez, and Enrique Onieva. An intelligent v2i-based traffic management system. *IEEE Transactions on Intelligent Transportation Systems*, 13(1):49–58, 2012. 1

[27] Aryan Mobiny, Pengyu Yuan, Supratik K Moulik, Naveen Garg, Carol C Wu, and Hien Van Nguyen. Dropconnect is effective in modeling uncertainty of bayesian deep networks. *Scientific reports*, 11(1):5458, 2021. 8

[28] Jay Nandy, Wynne Hsu, and Mong Li Lee. Towards maximizing the representation gap between in-domain & out-of-distribution examples. *NeurIPS*, 33:9239–9250, 2020. 8

[29] Radford M Neal. *Bayesian learning for neural networks*. Springer Science & Business Media, 2012. 8

[30] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *NeurIPS*, 32, 2019. 8

[31] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *NeurIPS*, 30, 2017. 4

[32] Rui Qian, Xin Lai, and Xirong Li. 3d object detection for autonomous driving: A survey. *Pattern Recognition*, 130: 108796, 2022. 1, 8

[33] Maithra Raghu, Katy Blumer, Rory Sayres, Ziad Obermeyer, Bobby Kleinberg, Sendhil Mullainathan, and Jon Kleinberg. Direct uncertainty prediction for medical second opinions. In *ICML*, pages 5281–5290. PMLR, 2019. 8

[34] Roopa Ravish and Shanta Ranga Swamy. Intelligent traffic management: A review of challenges, solutions, and future perspectives. *Transport and Telecommunication Journal*, 22 (2):163–182, 2021. 1

[35] Omer Sagi and Lior Rokach. Ensemble learning: A survey. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, 8(4):e1249, 2018. 8, 1

[36] Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. *NeurIPS*, 31, 2018. 8

[37] Divya Shanmugam, Davis Blalock, Guha Balakrishnan, and John Guttag. Better aggregation in test-time augmentation. In *ICCV*, pages 1214–1223, 2021. 8

[38] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *CVPR*, pages 770–779, 2019. 8

[39] 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 *CVPR*, pages 10529–10538, 2020. 8

[40] Weijing Shi and Raj Rajkumar. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *CVPR*, pages 1711–1719, 2020. 8

[41] Yingjie Wang, Qiuyu Mao, Hanqi Zhu, Jiajun Deng, Yu Zhang, Jianmin Ji, Houqiang Li, and Yanyong Zhang. Multi-modal 3d object detection in autonomous driving: a survey. *International Journal of Computer Vision*, 131(8):2122–2152, 2023. 1

[42] Florian Wenzel, Kevin Roth, Bastiaan S Veeling, Jakub Świątkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How good is the bayes posterior in deep neural networks really? *arXiv preprint arXiv:2002.02405*, 2020. 8

[43] Hai Wu, Shijia Zhao, Xun Huang, Chenglu Wen, Xin Li, and Cheng Wang. Commonsense prototype for outdoor unsupervised 3d object detection. In *CVPR*, 2024. 1, 7

[44] Hai Wu, Shijia Zhao, Xun Huang, Chenglu Wen, Xin Li, and Cheng Wang. Commonsense prototype for outdoor unsupervised 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14968–14977, 2024. 4

[45] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 8

[46] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Std: Sparse-to-dense 3d object detector for point cloud. In *ICCV*, pages 1951–1960, 2019. 8

[47] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *CVPR*, pages 11040–11048, 2020. 8

[48] Yuan Yao, Yuanhan Zhang, Zhenfei Yin, Jiebo Luo, Wanli Ouyang, and Xiaoshui Huang. 3d point cloud pre-training with knowledge distillation from 2d images. *arXiv preprint arXiv:2212.08974*, 2022. 7

[49] Junbo Yin, Dingfu Zhou, Liangjun Zhang, Jin Fang, Cheng-Zhong Xu, Jianbing Shen, and Wenguan Wang. Proposal-contrast: Unsupervised pre-training for lidar-based 3d object detection. In *ECCV*, pages 17–33. Springer, 2022. 7

[50] Yurong You, Katie Luo, Cheng Perng Phoo, Wei-Lun Chao, Wen Sun, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Learning to detect mobile objects from lidar scans without labels. In *CVPR*, pages 1130–1140, 2022. 1, 4, 5, 7, 8, 3

[51] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE access*, 8:58443–58469, 2020. 1

[52] Hu Zhang, Jianhua Xu, Tao Tang, Haiyang Sun, Xin Yu, Zi Huang, and Kaicheng Yu. Opensight: A simple open-vocabulary framework for lidar-based object detection. In *ECCV*, 2024. 1

[53] Lunjun Zhang, Anqi Joyce Yang, Yuwen Xiong, Sergio Casas, Bin Yang, Mengye Ren, and Raquel Urtasun. Towards unsupervised object detection from lidar point clouds. In *CVPR*, pages 9317–9328, 2023. 1, 5, 7, 8

[54] Ruiyang Zhang, Hu Zhang, Hang Yu, and Zhedong Zheng. Approaching outside: Scaling unsupervised 3d object detection from 2d scene. In *ECCV*, 2024. 1, 4, 5, 7

[55] Ruiyang Zhang, Hu Zhang, and Zhedong Zheng. Vl-uncertainty: Detecting hallucination in large vision-language model via uncertainty estimation. *arXiv:2411.11919*, 2024. 8

[56] Ruiyang Zhang, Hu Zhang, Hao Fei, and Zhedong Zheng. Uncertainty-o: One model-agnostic framework for unveiling uncertainty in large multimodal models. *arXiv:2506.07575*, 2025. 8

[57] Xuanmeng Zhang, Zhedong Zheng, Minyue Jiang, and Xiaoqing Ye. Self-ensembling depth completion via density-aware consistency. *Pattern Recognition*, 154:110618, 2024. 8

[58] Jingyuan Zhao, Wenyi Zhao, Bo Deng, Zhenghong Wang, Feng Zhang, Wenxiang Zheng, Wanke Cao, Jinrui Nan, Yubo Lian, and Andrew F Burke. Autonomous driving system: A comprehensive survey. *Expert Systems with Applications*, page 122836, 2023. 1

[59] Zhedong Zheng and Yi Yang. Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation. *International Journal of Computer Vision*, 129(4):1106–1120, 2021. 8

[60] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *CVPR*, pages 4490–4499, 2018. 8

Harnessing Uncertainty-aware Bounding Boxes for Unsupervised 3D Object Detection

Supplementary Material

6. More Discussions

Does detection backbone in UA3D only use a dense prediction head? No, UA3D does not alter the overall detection pipelines of the original 3D detection backbone. For example, in PointRCNN, the detection process still includes both a dense prediction head and an ROI head. However, the uncertainty estimation and regularization process are conducted during the dense prediction head for two reasons: (1) The number of dense predictions corresponds to the number of points in the point cloud, which remains consistent across different inferences on the same point cloud. This makes it convenient for the uncertainty estimation process, as primary and auxiliary dense predictions can be directly matched and compared. (2) Dense predictions cover the full prediction of the 3D detector, facilitating a more comprehensive uncertainty estimation process.

Can the discrepancy between primary and auxiliary detector predictions effectively capture uncertainty? Yes. For accurate pseudo-boxes (with low uncertainty) that match the distribution of object points, both detectors tend to generate similar detection results. Conversely, for inaccurate pseudo-boxes (with high uncertainty), one detector could produce accurate results based on knowledge learned from other training data, while the other can overfit to the inaccurate pseudo-boxes. Consequently, discrepancies in predictions can be observed, effectively capturing uncertainty.

Is UA3D limited to PointRCNN? No, UA3D is not limited to specific detection backbones. For 3D detectors with various structures, the detection process typically concludes with different detection heads. UA3D can achieve uncertainty estimation by duplicating an existing head to create primary and auxiliary detectors. The discrepancy between these two detectors' predictions can be utilized to estimate uncertainty and implement the regularization process. Based on this principle, in PointRCNN, we choose the dense head to perform fine-grained uncertainty estimation and regularization.

How is the auxiliary detector initialized? The auxiliary detector is trained from scratch. We do not rely on pre-trained checkpoints. The initialization step is the same as that of the original primary detector. This ensures generalizability across various 3D detector structures, as no specific or fixed design is adopted.

Why can uncertainty estimation reflect the inaccuracy of pseudo boxes? Accurate pseudo boxes are well-aligned with the object regions in the input point cloud, typically exhibiting consistent characteristics such as tightly enclosing specific point groups and maintaining a reasonable size. In contrast, inaccurate pseudo boxes show significant and unpredictable variations, making them harder to interpret. This inherent uncertainty can confuse the model, leading to highly varying predictions for the same object. Consequently, discrepancies between the two detector predictions indicate elevated uncertainty, reflecting the inaccuracy of pseudo boxes.

Why choose dense predictions for uncertainty estimation instead of using predictions from the Region-of-Interest (ROI) head? Since the dense outputs predict a box for each point in the point cloud, they generate the same number of predictions regardless of the model structure, ensuring consistency between primary and auxiliary detectors. This consistency naturally simplifies the calculation of differences between two detector predictions for estimate uncertainty. In 3D detection model, ROI head aggregates point-wise predictions into certain numbers of final bounding boxes, and the numbers of predicted boxes can vary between the primary and auxiliary detectors. While it is feasible to utilize the output from ROI head for uncertainty estimation, the different numbers of boxes from primary and auxiliary detectors require a matching process. Matching boxes between two detectors introduces significant computational overhead. Given the additional training cost, we choose not to rely on the predictions from ROI head.

Why is uncertainty regularization fine-grained? Our calculation process operates at the box coordinate level. This allows our method to identify coordinate-specific inaccuracies in pseudo boxes and dynamically mitigate their negative influence. During the pseudo box generation process, pseudo boxes can exhibit inaccuracies in specific coordinates, such as only in the orientation angle. In such cases, treating the entire box as fully certain or uncertain is not reasonable. Our fine-grained regularization approach can selectively reduce the negative influence of the inaccurate coordinate while preserving the efficacy of other accurate coordinates.

What differentiates our work from the model ensemble approaches [35]? We focus on improving the performance of a single model. Our final detection results benefit from regularization gained from both the primary and auxiliary detectors. During the inference phase, we only enable the primary detector, rather than typical model ensemble approaches that aggregate multiple different models. Notably, our approach is also scalable and can be applied to individual models within an ensemble, if desired.

Why not conduct experiment on Waymo? We choose datasets with multi-traversal data, which is essential for a fair comparison with existing method MODEST. Since Waymo does not contain multi-traversal data, we do not utilize this dataset.

Could two branches yield similar predictions for noisy pseudo boxes? Or could auxiliary branch introduce noise for accurate pseudo boxes? Those cases could happen, while as corner cases. To provide an overview of UA3D uncertainty estimation results, we present the statistical uncertainty distribution (see Fig. 2). We observe a clear gap between uncertainty distributions of accurate pseudo boxes and noisy ones. Overall, UA3D could not address 100% noisy cases. However, for most inaccurate pseudo labels, they are assigned with high uncertainty. UA3D mitigates negative influence of most noisy pseudo labels, and finally improves detector performance.

Why not utilize data augmentation to cause variance in predictions? Data augmentation-based methods are time-consuming as they require multiple inferences. In contrast, UA3D processes data with an auxiliary branch in a single forward pass, making uncertainty estimation more efficient.

Can uncertainty be pre-calculated, so that no calculation is needed during training? Pre-calculated pseudo label uncertainty like confidence score is good for initialization, but tends to degrade in quality as training progresses. For instance, certain samples that initially exhibit high uncertainty become increasingly reliable over the course of training. Therefore, UA3D adopt the on-the-fly uncertainty, which surpass the pre-defined uncertainty (see Tab. 3).

Does UA3D have tendency to predict high uncertainty? We add uncertainty U into loss to suppress this tendency. Losses for two detectors are $\mathcal{L}_p^u = \sum_{i=1}^{7}(\frac{\mathcal{L}_{p,i}}{\exp(\boldsymbol{U_i})} + \lambda \cdot \boldsymbol{U_i})$, and $\mathcal{L}_a^u = \sum_{i=1}^{7}(\frac{\mathcal{L}_{a,i}}{\exp(\boldsymbol{U_i})} + \lambda \cdot \boldsymbol{U_i})$ (see Eq. 2). The $\lambda \cdot \boldsymbol{U_i}$ serves as penalty term for consistently high uncertainty.

Can UA3D improve detector recall? UA3D does improve both precision and recall. Noisy or inaccurate labels are given less weight, while all accurate labels keep their weights. This means reliable labels naturally get more emphasis within every iteration. By focusing more on these accurate labels, UA3D not only improves precision but also helps increase recall (see Fig.5 (b)).

Why not apply different augmentations to the input point cloud for the primary and auxiliary detectors to better capture uncertainty? Different perturbations in the input point cloud could enhance the uncertainty estimation process. However, we have observed that the proposed primary and auxiliary detector design is already sufficient to capture uncertainty. Therefore, we do not adopt additional point cloud augmentation.

Can UA3D improve fully supervised training processes? Yes, UA3D can enhance training using human labels. Even annotations from human experts can contain inaccuracies and noise, due to the inherent difficulty in annotating precise 3D boxes for distinct objects. UA3D can mitigate the negative impact of such noisy labels and potentially improve model performance. However, the issue of inaccurate pseudo-boxes is more severe in unsupervised settings. Therefore, we focus on this setting to better demonstrate the effectiveness of UA3D.

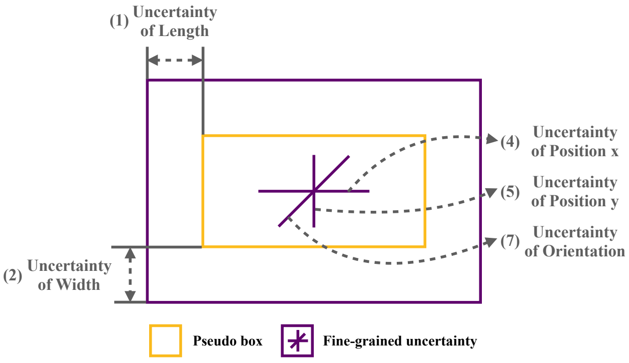

(1) Uncertainty of Length
(4) Uncertainty of Position x
(5) Uncertainty of Position y
(7) Uncertainty of Orientation
(2) Uncertainty of Width
Pseudo box
Fine-grained uncertainty

Figure 6. **Detailed explanation of our uncertainty visualization in Bird's Eye View (BEV).** **(1)** Uncertainty of length: it is visualized by the gap between the length coordinates of the **purple** and yellow boxes. **(2)** Uncertainty of width: it is similarly represented by the gap between the width coordinates of the two boxes. **(3)** Uncertainty of height: it is depicted as the gap between the height coordinates of the two boxes, though it is omitted in BEV for brevity. **(4)** Uncertainty of position x: it is shown by the length of the **purple** line extending horizontally (left-to-right). **(5)** Uncertainty of position y: it is represented by the length of the **purple** line extending vertically (top-to-bottom). **(6)** Uncertainty of position z: it is visualized by the length of the **purple** line along the z-axis, but it is not shown in BEV for simplicity. **(7)** Uncertainty of orientation: it is denoted by the length of the **purple** diagonal line.

7. Explanation of Uncertainty Visualization

Here we first elaborate our uncertainty visualization in Fig. 6. The uncertainties in length, width, and height are represented by the gap between the corresponding coordinates of the **purple** and yellow boxes. For the uncertainties in position (x, y, z) and orientation, they are visualized by the lengths of the **purple** lines along the respective directions.

8. More Qualitative Results

Detection Results Comparison. We present additional qualitative results in Fig. 7. As shown in Fig. 7 (a), our uncertainty-aware framework generates more accurate predictions regarding object shape, location, and orientation. This improvement is attributed to our proposed uncertainty estimation and regularization, which mitigate the negative effects of inaccurate pseudo boxes at a fine-grained coordinate level. Fig. 7 (b) further shows that our method is more effective in recalling difficult object categories, *e.g.*, far and small objects. Our uncertainty-aware framework enhances the prominence of accurate pseudo boxes for these challenging objects, facilitating more effective recognition of those objects.

Correspondence Between Noisy Pseudo Box and High Uncertainty. We further present a detailed analysis for the correspondence between noisy pseudo box and high estimated uncertainty (see Fig. 8).

9. Implementation Details

Hyper-parameters. We follow MODEST [50] settings. for nuScenes [2], the batch size is set to 2 per GPU. We conduct training for 80 epochs using the Adam optimizer with a one-cycle policy. The initial learning rate is 0.01, with a weight decay of 0.01 and a momentum of 0.9. Learning rate decay is applied at epochs 35 and 45 with a decay rate of 0.1. Additionally, a learning rate clip of $1e^{-7}$ and a gradient norm clip of 10 are employed. We perform one round of seed training followed by 10 rounds of self-training for all experiments. Each round of training takes approximately 4 hours, resulting in a total training time of about 44 hours (4 hours × 11 rounds). For Lyft [11], we reduce the number of epochs to 60 for efficiency, considering that the Lyft dataset is 3 times larger than nuScenes. The self-training pipeline for Lyft also consists of one round of seed training and 10 rounds of self-training. Each training round takes approximately 12 hours, leading to a total training time of around 131 hours (12 hours × 11 rounds). Other settings remain the same as those for nuScenes, without specific tuning, to validate the generalizability of our proposed uncertainty-aware framework.

Data Processing. For both nuScenes and Lyft, we apply several data augmentations. We sample 6,144 points per point cloud for nuScenes, while for Lyft, we sample 12,288 points per point cloud, as the point clouds in Lyft are generally denser than those in nuScenes. We perform random world flipping of the entire point cloud along the x-axis. We also apply random world rotation within the angle range of [-0.785, 0.785] and random world scaling within the scale ratio range of [0.95, 1.05]. Point shuffling is applied to the training set but not to the test set. We focus on object discovery, following the trajectory of previous works such as MODEST, OYSTER, and LiSe. We do not explicitly consider object categories during the experiments.

Self-training Pipeline. Our uncertainty-aware framework operates within a self-training pipeline, following the common settings in previous works [50]. In general, a self-training pipeline consists of two stages: seed training and self-training. Initial generated pseudo boxes are referred to as seeds. During the seed training, an initial detection model is trained based on those seeds. Then the trained model from previous round is first applied to the training set to obtain refined pseudo boxes. During the self-training, a new detection model is trained on the refined pseudo boxes. The process is iteratively repeated for T rounds.

We visualize the obtained uncertainty in Fig. 8 and such analysis further validates the correspondence between the pseudo boxes inaccuracies and estimated uncertainty. Specifically, we observe that accurate pseudo boxes, which typically lead to consistent predictions from both the primary and auxiliary detectors, exhibit low uncertainty. In contrast, when a pseudo box shows inaccuracies in certain coordinates, the estimated uncertainty for those coordinates is significantly higher since the predictions from the primary and auxiliary detectors diverge on those coordinates.

10. Real-world Application and Limitations

Application. There are several potential ways in which unsupervised 3D object detection could benefit real-world applications. The unsupervised setting enables large-scale pretraining on vast amounts of unlabeled data. Additionally, the generated pseudo labels can serve as initial raw annotations, which can then be refined through human filtering, thereby reducing annotation costs.

Limitations. We provide a statistical overview of our estimated uncertainty in Fig. 2. We observe that most inaccurate pseudo boxes are assigned with high uncertainty. However, a few cases with incorrectly estimated uncertainty cannot be fully avoided in our framework and our proposed method tends to fall short in addressing these cases.

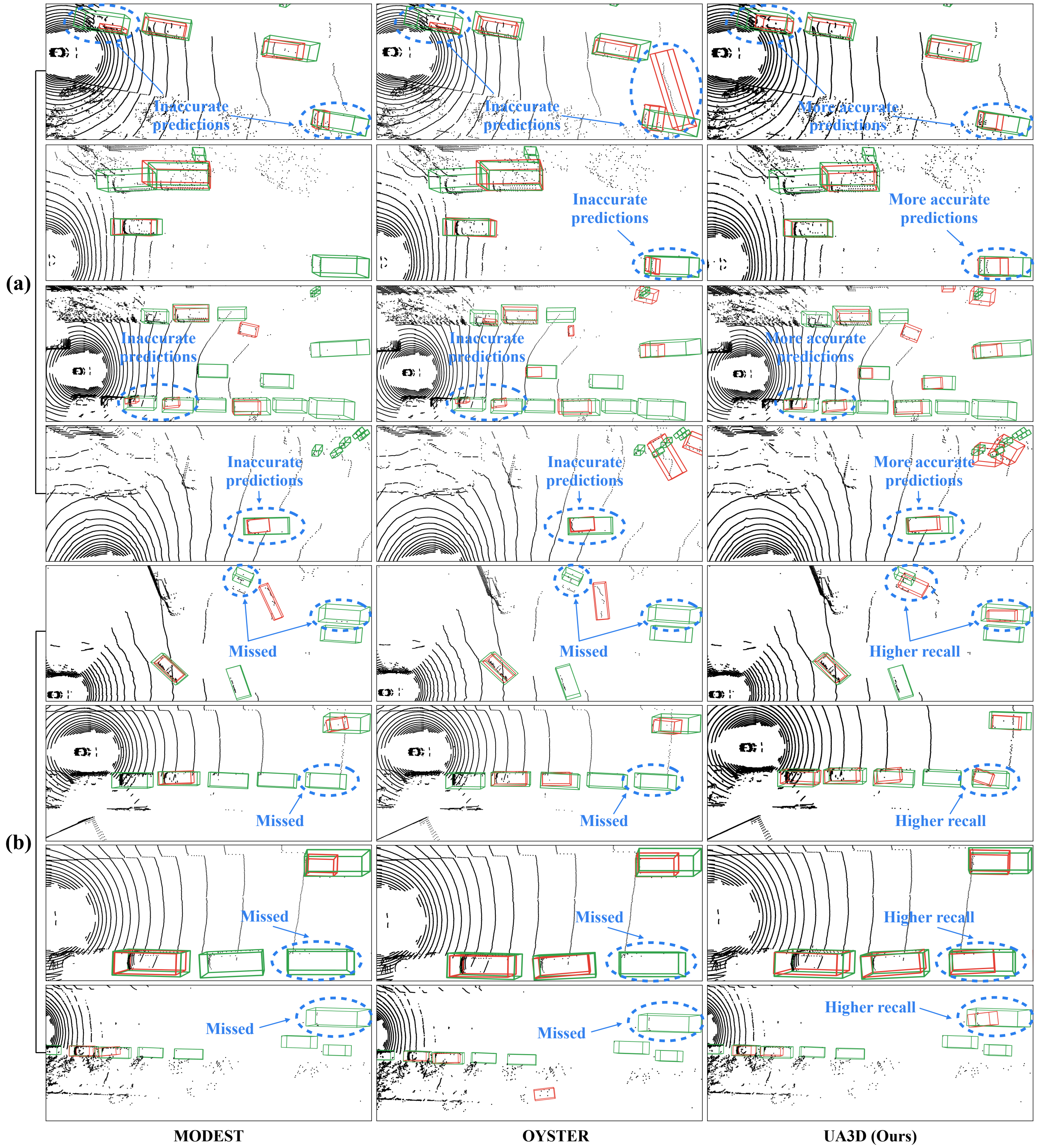


Figure 7. **Further qualitative comparison between different methods.** We compare our uncertainty-aware framework with previous works, *e.g.*, MODEST and OYSTER. **Green** boxes denote the ground-truth and **red** boxes represent predictions from the detection model. (a) Our uncertainty-aware framework shows more accurate perceptions of various foreground objects. (b) In challenging scenarios, such as distant objects with sparse point clouds or small objects, our method achieves a higher recall rate.

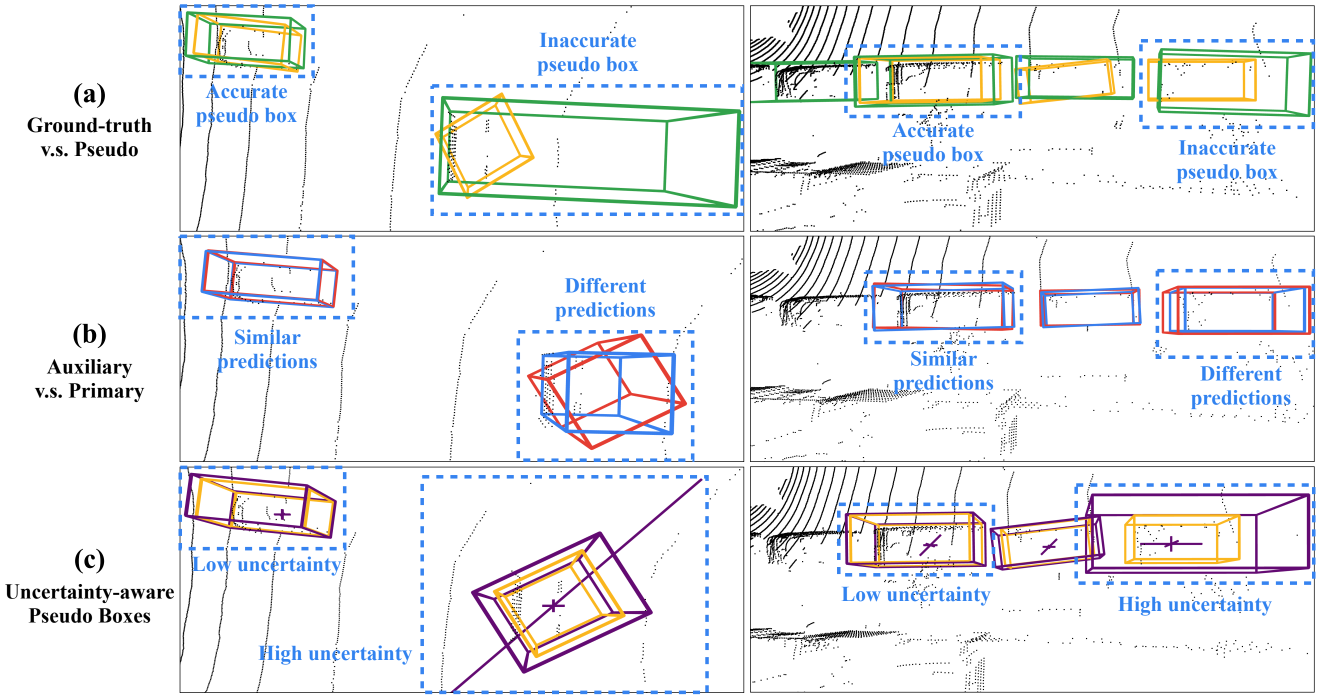


Figure 8. **Correspondence between pseudo label inaccuracy and high uncertainty.** (a) We present ground truth and pseudo boxes in two different point clouds (left and right columns). Each point cloud contains both accurate and inaccurate pseudo boxes. We observe that pseudo boxes can be significantly inaccurate in terms of the shape, location, and rotation. Direct usage of these boxes for training can easily impair the performance of the detection model. (b) We present the predictions from the primary and auxiliary detectors. Two detector predictions align closely for objects with accurate pseudo boxes but diverge for those with inaccurate ones. The mismatch between inaccurate pseudo boxes and the actual point cloud distribution can confuse the model, resulting in varying interpretations. (c) We present our uncertainty-aware pseudo boxes. Fine-grained coordinate-level uncertainty is estimated, *e.g.*, the orientation uncertainty for the right object (in left column) is high (as indicated by the long **purple diagonal line**), due to its inaccuracy in the pseudo box. The *colors* follow the same conventions in Fig. 3.