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 pretrained 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.

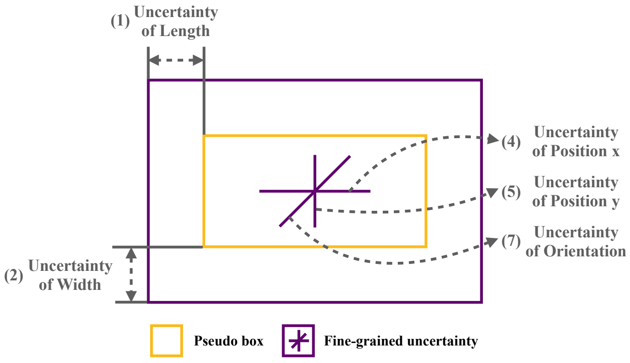
(1) Uncertainty of Length
(4) Uncertainty of Position x
(5) Uncertainty of Position y
(7) Uncertainty of Orientation
(2) Uncertainty of Width
Pseudo box
Fine-grained uncertainty

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.

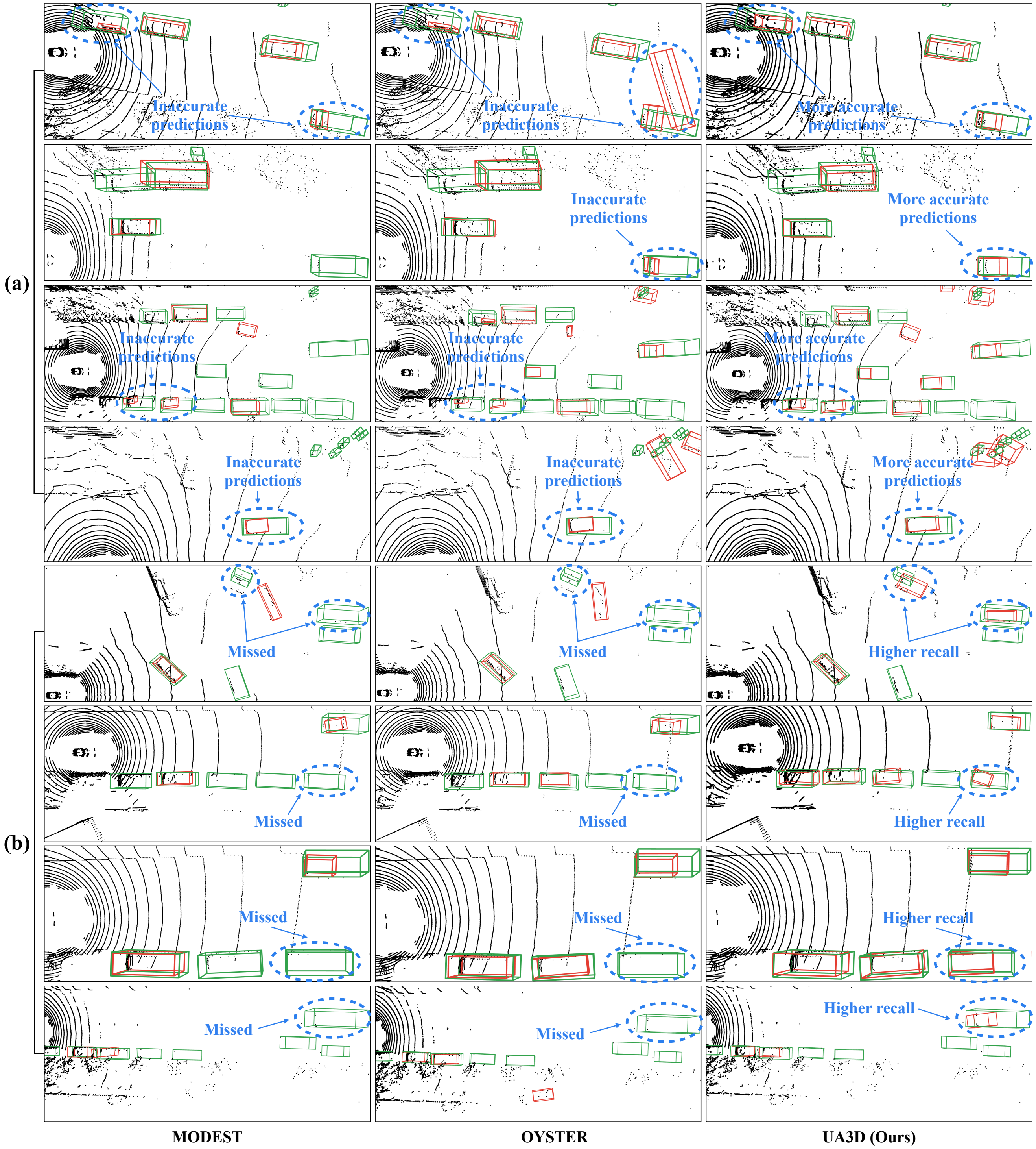

Figure 7. **Further qualitative comparison between different methods.** We compare our uncertainty-aware framework with previous works, *e.g.*, MODEST and OYSTER. **Green** boxes denote the ground-truth and **red** boxes represent predictions from the detection model. (a) Our uncertainty-aware framework 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.

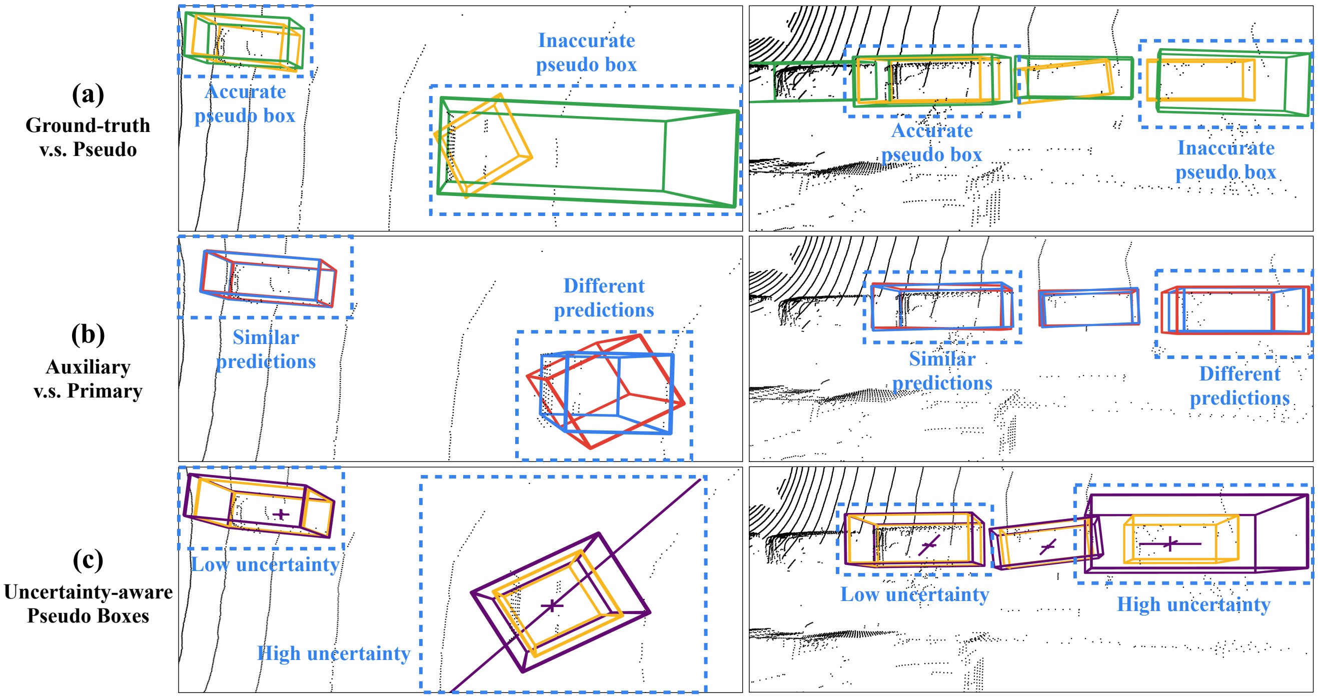

Figure 8. **Correspondence between pseudo label inaccuracy and high uncertainty.** (a) We present ground truth and pseudo boxes in two different point clouds (left and right columns). Each point cloud contains both accurate and inaccurate pseudo boxes. We observe that pseudo boxes can be significantly 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.