A. Appendix

A.1. Datasets and Metrics

KITTI-3D. KITTI [21] is the most popular dataset in autonomous driving scenario in the past decade. KITTI-3D is a subset of KITTI dataset, which contains 7,481 and 7,518 frames for training and testing respectively. In each frame, the stereo images, synchronized LiDAR sweep, calibration files, and 3D bounding box annotations are provided. Following [11, 12], we split the training frames into a training set (3,712 frames) and a validation set (3,769 frames), and conduct the experiments in this split. Following [70], we adopt the AP_{40} of 3D detection and Bird's Eye View (BEV) detection as metrics. We mainly focus on the Car category on KITTI-3D in the main paper, and both 0.7 and 0.5 IoU thresholds are considered. In this supplementary, we also discuss the Pedestrian and Cyclist categories with 0.5 IoU threshold.

nuScenes. nuScenes [6] is a large-scale autonomous driving dataset proposed in 2020. It provides about 40K annotated frames of 10 categories for the panoramic view in the autonomous driving scenario. In particular, there are 28,130 frames, 6,019 frames, and 6,008 frames for training, validation, and testing respectively (six images per frame). As for evaluation metrics, nuScenes uses a 2D center error (in the ground plane) based AP metric. Specifically, they consider four thresholds, $\mathbb{D} = \{0.5, 1, 2, 4\}$, for AP computing, and average them over 10 categories to get the final mAP:

$$\mathrm{mAP} = \frac{1}{|\mathbb{C}||\mathbb{D}|} \sum_{c \in \mathbb{C}} \sum_{d \in \mathbb{D}} \mathrm{AP}_{c,d}, \tag{3}$$

where $|\mathbb{C}|$ and $|\mathbb{D}|$ are the category set and threshold set respectively. Furthermore, five true positive metrics (TP metrics) are also considered, including Average Translation Error (ATE), Average Scale Error (ASE), Average Orientation Error (AOE), Average Velocity Error (AVE), and Average Attribute Error (AAE). For each TP metric, the mean TP metric (mTP metric) are computed by:

$$\mathrm{mTP}_k = \frac{1}{|\mathbb{C}|} \sum_{c \in \mathbb{C}} \mathrm{TP}_{k,c}, \tag{4}$$

where k is the index of TP metrics, *e.g.* TP_1 is ATE. Finally, the nuScenes detection score (NDS) is computed by weighted averaging the above metrics:

$$\mathrm{NDS} = \frac{1}{10}[5 \cdot \mathrm{mAP} + \sum_{k=1}^{5}(1 - \min(1, \mathrm{mTP}_k))]. \tag{5}$$

ScanNet V2. ScanNet V2 [17] is a richly annotated dataset of 3D reconstructed of indoor scenes, and 3D box annotations can be derived from the annotated meshes [58]. There are about 1.2K training samples with 18 object different categories for hundreds of rooms. In this dataset, we adopt the 3D IoU based mAP as metric, and 0.25 and 0.5 IoU thresholds are considered in model evaluation and error analysis.

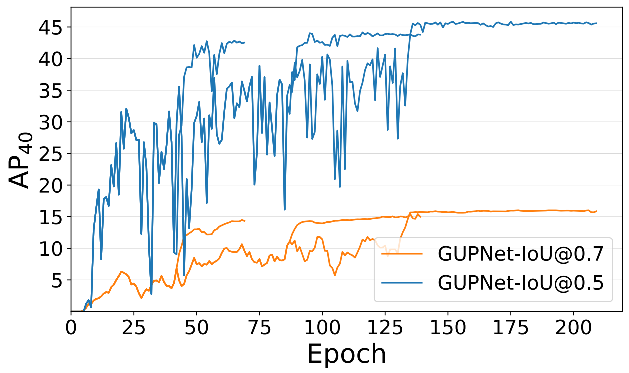


Figure 4: Learning curves on KITTI-3D without smoothing.

A.2. Average Height Error

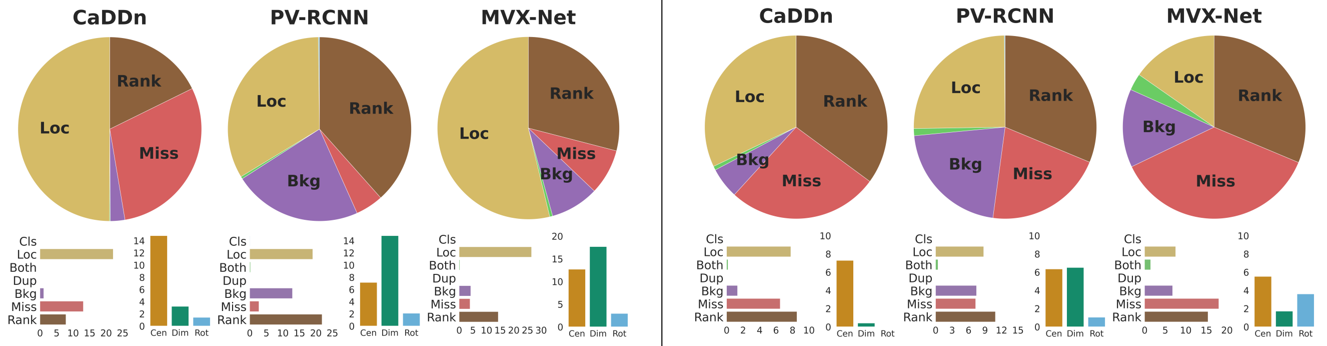
In the main paper, Table 8 shows that different evaluation metrics have their preferences for models. Specifically, on the nuScenes dataset, FCOS3D and BEVDet show similar performances under the nuScenes metric, but there is a large gap under the KITTI metric. To further explore this phenomenon, we define Average Height Error (AHE) as the 1D Euclidean distance of the center height of the bounding box, in order to bridge the gap between 2D IoU and 3D IoU. Table 9 shows that the predicted center height of BEVDet is significantly worse than that of FCOS3D. It reflects that the nuScenes metric is more friendly to BEV-based methods.

	mATE	mAHE
FCOS3D	0.78	0.11
BEVDet	0.72	0.15

Table 9: Average Translation Error (ATE) and Average Height Error (AHE) of BEVDet and FCOS3D for all categories on nuScenes *validation* set.

A.3. Implementation of TIDE3D

In the main paper, we introduce the design principles of TIDE3D, and there are some minor implementation differences on different datasets. Specifically, in KITTI-3D, TIDE3D conduct error diagnosis for three categories with three different settings separately. We directly compute the 3D IoU of detections and ground truths in the 3D space, instead of computing the BEV IoU and multiplying it by the 1D IoU at the height dimension. This design allows us to get more accurate 3D IoU for the objects whose roll angles and pitch angles are not zero (the roll angles and pitch angles of objects are always zero in KITTI-3D, but not nec-



(a) Pedestrian (b) Cyclist

Figure 5: **Example error diagnosis results.** We show the results of various models for the Pedestrian category and Cyclist category on KITTI-3D *validation* set under the moderate setting with 0.5 IoU threshold. Specifically, we select CaDDn [63] (image-based), PV-RCNN [67] (LiDAR-based) and MVX-Net [72] (fusion-based) to explore the bottleneck of detectors with different modalities.

essarily for other datasets, such as nuScenes). In ScanNet V2, the error distribution is collected over all categories, *i.e.* based on mAP. For nuScene, we compute both ΔmAP and ΔNDS for error diagnosis (we mainly focus on ΔmAP). Besides, there is a hyper-parameter t_f in TIDE3D, which is used to identify whether a false positive belongs to 'localization error' or 'background error'. We set $t_f = 0.1$ for 3D IoU based metrics and $t_f = 5$ for center-distance based metrics.

A.4. Training Curve

Figure 4 shows the un-smoothed learning curve of GUP-Net in KITTI-3D (Figure 1, right), and we can see that the performance fluctuates violently during training. Even in the later stage of training, the AP_{40} is still unstable (especially for the $1\times$ schedule). Meanwhile, we also observe the final performance fluctuates over multiple runs (similar to Table 20 in DEVIANT [29]), and reporting mean values of multiple runs is recommended for future work.

A.5. More TIDE3D Analysis

Pedestrian and Cyclist. Here we show the error diagnosis result and discussion for the Pedestrian and Cyclist categories in Figure 5. We can find that the major issue of the image-based model (CaDDn) is still the inaccurate localization. For the LiDAR-based model (PV-RCNN), the localization error is relatively small and mainly caused by the inaccurate estimation of dimension (instead of location). This is probably because LiDAR provides accurate location information but can only capture the surface of the objects. Besides, different from image-based methods, we can find that background error accounts for a large part of the whole error distribution. This is because LiDAR points lack color information, and the models base on them easily generate false positives when they meet objects with similar structures to the objects of interest. See Figure 6 for a typical detection result.

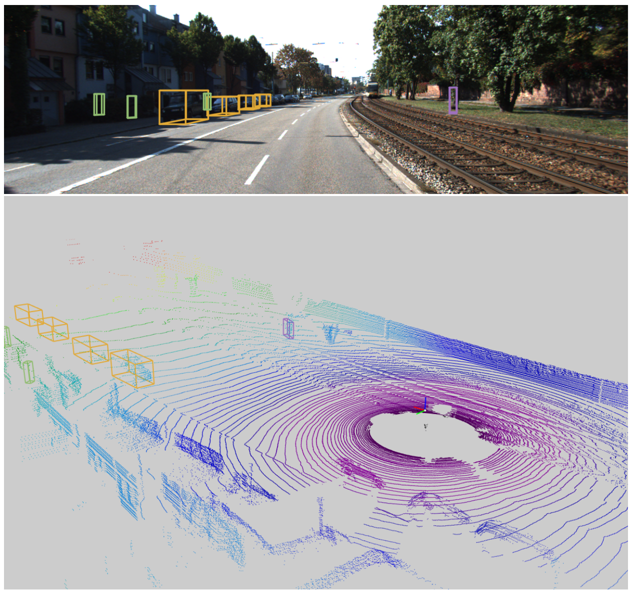


Figure 6: An typical error case for LiDAR-based methods. A tree trunk is wrongly recognized as a cyclist due to the lack of color information. Cars, cyclists, and pedestrians are visualized with yellow, purple, and green boxes. Figures are copied from [48].

Error diagnosis in ScanNet V2. In addition to autonomous driving scenario, TIDE3D can also be used in indoor scenes. Here we use ScanNet v2 and VoteNet [58], which is a powerful detection model based on point cloud, as an example. Figure 7 shows that VoteNet has different error distributions

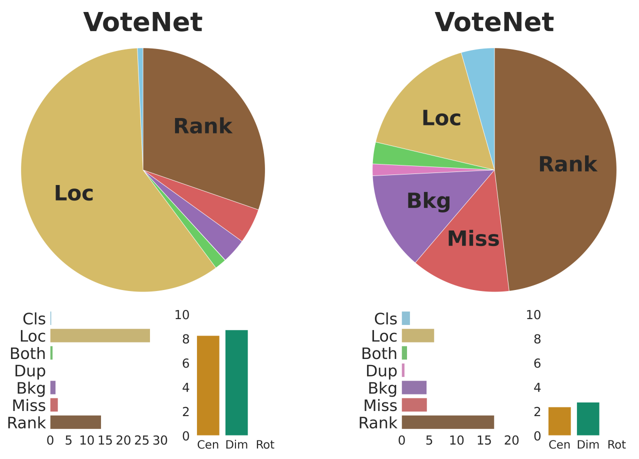


Figure 7: **Example error diagnosis results of VoteNet on ScanNet V2 *validation* set.** We show the results with 0.5 IoU threshold (*left*) and 0.25 IoU threshold (*right*) for all categories.

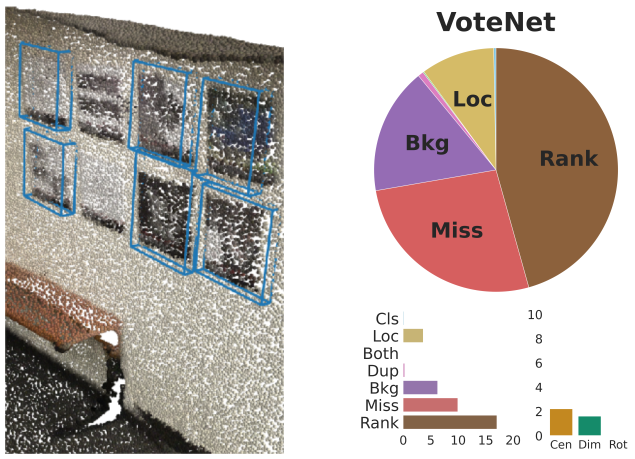


Figure 8: **Error diagnosis of VoteNet for 'picture' category.** The error distribution is collected with the IoU threshold of 0.25 on ScanNet V2. A training sample with several 'picture' items is provided to show the low quality of annotation, which is a major reason leading to the high background and missing errors. We use blue boxes to denote the ground truth items. Best viewed in color with zoom-in.

under different IoU thresholds. Specifically, under a strict IoU threshold of 0.5, the model error is dominated by localization error which is caused by the inaccurate estimation of center and dimension (note the bounding box annotation in ScanNet V2 is axis-aligned and the model does not need to predict the rotation angle). For a loose threshold of 0.25, the localization error is mitigated and the overall performance is limited by more factors, including ranking error, missing error, or background error.

In order to analyze the diverse problems encountered in detecting different categories of objects, we can further analyze the AP of a specified single category. As an example, we look at the **'picture'** which is the worst performing category as shown in Figure 8. Since the pictures usually have small sizes and easily blend into the background point cloud of the wall, they are quite unrecognizable especially when there is no color information provided by the image data. Besides, incomplete annotations also confuse the detector's recognition of the pictures. Therefore, the main problems besides ranking error are missing error and background error for this category.

A.6. Discussion about Training Recipes

We observe the common choices of training recipes show varying impacts on different models. Here we give some notes on model training:

i. The optimal training recipes are different for different models. For example, the detection models with deformable convolutions [96], such as [44, 92], require a lower learning rate (*e.g.* $1.25e^{-4}$) for stable training, while other models with similar network architecture [51, 46] perform better at a larger one (*e.g.* $1.25e^{-3}$).

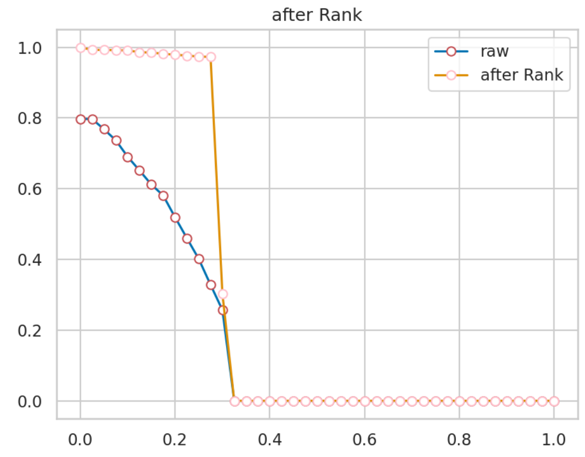
ii. Some data augmentations may have similar effects. For example, applying random crop (at a small-scale) and random shift separately can improve the accuracy, while applying both of them still get similar performance. This suggests that these two operations may work in a similar way, *i.e.* applying geometric transformations to the images, thus making the models sensitive to the location of objects.

iii. For different datasets/metrics, the same augmentation may have different effects. For example, random crop works well in KITTI-3D dataset, while the performance change of applying this operation on nuScene is relatively inconspicuous. This is mainly caused by the difference in data size and evaluation metrics, and the custom training recipes for different application scenarios are required.

iv. Data augmentation is an effective way to improve the performance of detection models, even in large-scale datasets [6, 75]. However, it is difficult to align geometric changes in the 2D image plane and the 3D world space, resulting in limited augmentation strategies. The popular BEV pipeline allow us to conduct data augmentation in the BEV space, which is a key factor for the success of such methods. Meanwhile, this also indicates that the algorithm performance can be improved by designing data augmentation strategies in the future.

A.7. Ranking Error in TIDE3D

We have shown that the ranking error, which is ignored in TIDE, is a major error type in 3D object detection systems. Different from other error types, ranking error in-



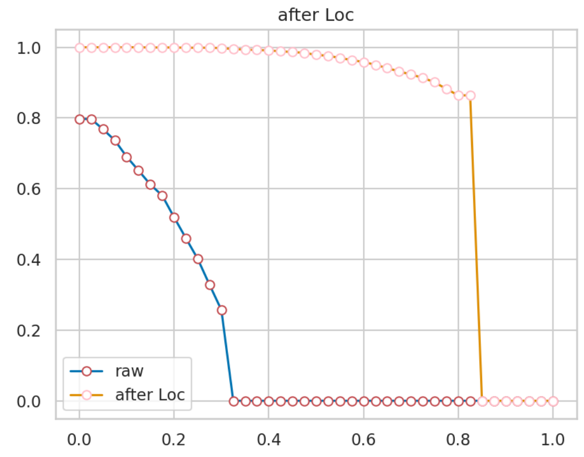


Figure 9: **PR curves.** We show the PR curves before/after applying the ranking oracle (**left**) and the localization oracle (**right**). The baseline is GUPNet and the metric is AP_{40} on KITTI-3D *validation* set under the moderate setting.

volves multiple predictions, and fixing this type may affect other error types. For example, in Section 6.3, we show the effects of 3D confidence with TIDE3D and find that both the localization error and ranking error are significantly reduced (similar phenomenon is shown in Table 2 of TIDE [2]), which is caused by the complicated interactions between multiple predictions. We argue that, although improving the quality of localization and confidence may show similar numbers in localization error, they achieve this in a completely different way and can be further identified. Specifically, we present the Precision-Recall (PR) curves before/after applying localization and ranking oracles in Figure 9. We can find that the ranking (misalignment) oracle improves the AP by maximizing the precision at each recall level, while the localization oracle corrects the false positive to true positive. All the figures we presented will be given from TIDE3D.