Supplementary Material for *"CAT-Det: Contrastively Augmented Transformer for Multi-modal 3D Object Detection"*

In this supplementary document, we provide more implementation details of the proposed CAT-Det framework in Sec. A as well as more ablation results on the main components of our approach in Sec. B including: the CMT module, the memory bank size, the trade-off parameter λ balancing the detection loss $(\mathcal{L}_{rpn} + \mathcal{L}_{rcnn})$ and contrastive learning loss $(\mathcal{L}_{cl-p}+\mathcal{L}_{cl-o})$, the GT-Paste and the ITB and PTB blocks. The runtime and complexity analysis of CAT-Det are also reported in Sec. B. In addition, we demonstrate more visualization results of CAT-Det on KITTI val split in Sec. C, and show detailed scores on the official KITTI test leaderboard in Sec. D.

A. More Implementation Details

In this section, we describe the details of the major parts in CAT-Det (as depicted in Fig. 2 in the main paper), including pointformer, imageformer, one-way multi-modal data augmentation (OMDA), and 3D box generation and point segmentation, together with the training losses.

The detailed network architecture is illustrated in Fig. A, and we elaborate each critical component as follows.

As displayed in Fig. 2 and Fig. 3 from the main paper, **Pointformer** consists of four point transformer blocks (PTBs), where the radii for ball query in PTBs are set to $[0.1, 0.5, 1.0, 2.0]$ and the channel sizes are fixed as $[96, 256, 512, 1024]$, respectively. The linear projection dimension in basic point transformer (BT) is set to 512. Similar to PointNet++, four feature propagation (FP) layers are adopted after stacked PTBs for up-sampling the point-cloud back to the original size with a stride of 4.

As shown in Fig. 4, **Imageformer** is composed of four image transformer blocks (ITBs), where the channel sizes are $[64, 128, 256, 512]$, respectively. For basic transformer, we use 4 self attention heads, of which the linear projection dimensions are fixed as 1024. The sizes of the input feature maps are $[640 \times 192, 320 \times 96, 160 \times 48, 80 \times 24]$ and those of the patches are $[32, 16, 8, 4]$. Similarly, following the cascaded ITBs, four up-sampling (UP) layers are employed to recover the image resolution with strides 2, 4, 8, 16, generating feature maps with the same size as the original image.

As for **OMDA**, similar to [4], we first generate a set of object-level point-clouds by cropping the points from the ground truth bounding boxes in the training data. Thereafter, we randomly select a subset of object-level point-clouds and paste them to a given LiDAR frame. With regard to contrastive learning, the temperature parameter τ is empirically fixed as 0.07. As to the memory bank, the momentum update hyper-parameter m is set to 0.999.

In **3D Box Generation and Point Segmentation as well as Training Losses**, as displayed in Fig. 2 from the main paper, we follow the existing work [2] and introduce point segmentation as an auxiliary task by employing an extra segmentation head $H_{seg}(\cdot)$, which is trained by the segmentation loss $\mathcal{L}_{seg}$. Due to space limit, we omit the description on $H_{seg}(\cdot)$ and $\mathcal{L}_{seg}$ in the main paper for succinctness. In this document, we provide more details.

Specifically, $H_{seg}(\cdot)$ consists of two fully connected (FC) layers, which is trained by the segmentation loss formulated as below:

$$\mathcal{L}_{seg} = \sum_{\boldsymbol{p}_i} \mathcal{L}_{focal}(\boldsymbol{p}_i), \tag{1}$$

where

$$\mathcal{L}_{focal}(\boldsymbol{p}_i) = -\alpha(1-p')^{\gamma}\log(p') \tag{2}$$

is the focal loss [1], and $\boldsymbol{p}_i$ is the i-th point. p' equals to p if $\boldsymbol{p}_i$ is the foreground point, and equals to $1-p$ otherwise, where p is the predicted confidence score. During training, we keep the default setting, *i.e.* $\alpha = 0.25$ and $\gamma = 2$ in Eq. (2). Note that the ground-truth segmentation mask is naturally provided by the labels, *i.e.* 3D points inside ground truth 3D boxes are considered as foreground points.

With point segmentation, a box regression head is introduced to generate 3D bounding box proposals. The feature for each proposal is obtained by randomly selecting 512 points in the corresponding proposal on top of the last layer of our two-stream multi-modal transformer. Subsequently, the refinement network consisting of three set abstraction (SA) layers is adopted to build a global representation, following which two cascaded 1×1 convolution layers for classification and regression are used to generate the prediction for detection including a 3D bounding box $(x, y, z, h, w, l, \theta)$ and a class confidence score c. Here, (x, y, z) indicates the 3D coordinate of the object center,

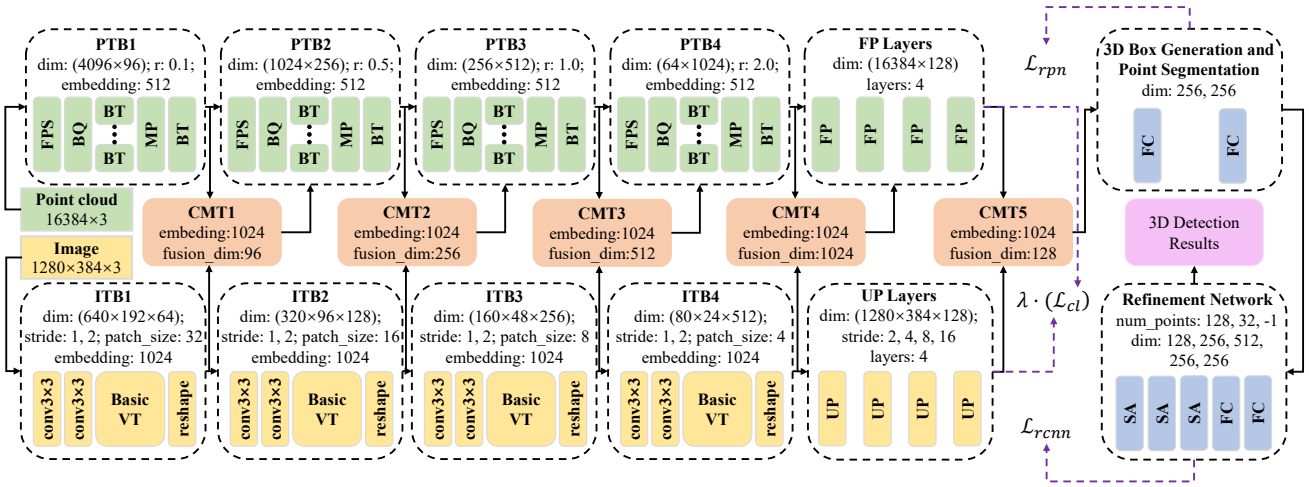

Figure A. Detailed network architecture of the proposed CAT-Det framework. PTB: the point transformer block; ITB: the image transformer block; CMT: the cross-modal transformer; FPS: the farthest point sampling; BQ: the ball query; BT: the basic point transformer; VT: the basic vision transformer; MP: the max-pooling; UP: the up-sampling layer; FP: the feature propagation layer; SA: the set abstraction layer; and FC: the fully connected layer.

(h, w, l) refers to the bounding box size, and θ is the orientation from the bird's eye view. $\mathcal{L}_{rpn}$ includes the point-cloud segmentation loss $\mathcal{L}_{seg}$ and the proposal generation loss $\mathcal{L}_{pg}$, *i.e.* $\mathcal{L}_{rpn} = \mathcal{L}_{seg} + \mathcal{L}_{pg}$. $\mathcal{L}_{pg}$ and $\mathcal{L}_{rcnn}$ denote the training objectives for the 3D proposal generation and refinement network, both of which consist of a classification loss and a regression loss. Concretely, for (z, h, w, l), we directly utilize the smooth $L1$ loss for regression. For (x, y, θ), we use the bin-based loss [2]. The overall 3D bounding box regression loss for the $i-$th bounding box is formulated as below:

$$\begin{aligned}
\mathcal{L}_{\text{bin}}^{(i)} &= \sum_{u \in \{(x,y,\theta)\}} (\mathcal{L}_{\text{ce}}(\widehat{\text{bin}}_u^{(i)}, \text{bin}_u^{(i)}) \\
&\quad + \mathcal{L}_{\text{smooth-}L1}(\widehat{\text{res}}_u^{(i)}, \text{res}_u^{(i)})), \\
\mathcal{L}_{\text{res}}^{(i)} &= \sum_{v \in \{(z,h,w,l)\}} \mathcal{L}_{\text{smooth-}L1}(\widehat{\text{res}}_v^{(i)}, \text{res}_v^{(i)}), \\
\mathcal{L}_{\text{box}}^{(i)} &= \mathcal{L}_{\text{bin}}^{(i)} + \mathcal{L}_{\text{res}}^{(i)},
\end{aligned} \tag{3}$$

where $\widehat{\text{bin}}^{(i)}$ and $\widehat{\text{res}}^{(i)}$ are the predicted bin assignments and residuals, respectively. $\text{bin}^{(i)}$ and $\text{res}^{(i)}$ are the ground-truth targets, $\mathcal{L}_{\text{ce}}$ and $\mathcal{L}_{\text{smooth-}L1}$ denote the cross-entropy classification loss and the smooth-$L1$ loss, respectively. Based on Eq. (3), $\mathcal{L}_{rcnn}$ is written as the following:

$$\mathcal{L}_{rcnn} = \frac{1}{\|\text{B}\|} \sum_{i \in \text{B}} \mathcal{L}_{ce}(prob_i, label_i) + \frac{1}{\|\text{B}_{pos}\|} \sum_{i \in \text{B}_{pos}} \mathcal{L}_{\text{box}}^{(i)} \tag{4}$$

where B is the set of 3D proposals from RPN and B_{pos} is the set of positive proposals for regression. $prob_i$ is the confidence score and $label_i$ refers to the corresponding ground-truth label. $\mathcal{L}_{pg}$ has the similar formulation as $\mathcal{L}_{rcnn}$.

B. More Ablation Results

In this section, we provide more ablation results w.r.t. the CMT module, the memory bank size and the trade-off-parameter λ.

As for **CMT**, we adopt this module after each PTB and ITB in four distinct levels as in Fig. 2 of the main paper, denoted by Layer-1, Layer-2, Layer-3, and Layer-4, respectively. We also adopt it between FPs in Pointformer and UPs in Imageformer, denoted as Layer-5. To validate the benefit of fully using CMT in all layers, we perform the ablation study by separately removing CMT from each layer. As summarized in Table A, removing CMT at an arbitrary level deteriorates the performance, and CMT plays a more important role in higher levels based on the observation that the performance decreases more sharply when they are removed in Layer-3∼5 than that in Layer-1∼2. The results in Table A also suggest that CMT can integrate multi-modal information in different levels, thus reaching the best performance when being fully used.

We further explore the effect of the **memory bank size** on the performance of OMDA, by varying it from 256 to 4,096. It is worth noting that the bank size determines the number of negative samples used. As shown in Table B, the mAP of CAT-Det increases as the bank size becomes larger, and reach the highest one when the size is 1,024. The reason behind lies in that the performance of OMDA increases when properly using more negative samples, but will be deteriorated when using too much negative pairs, since it probably leads to severe imbalance of positive/negative samples. We empirically set it to 1,024 in all the experiments.

Method	3D Object Detection (%)			
Levels	Car	Ped.	Cyc.	mAP
Fully used in Layer-1$\sim$ 5	**83.58**	**66.45**	**76.22**	**75.42**
Removed in Layer-1	83.35	66.18	75.93	75.15
Removed in Layer-2	83.21	65.83	75.54	74.86
Removed in Layer-3	82.84	65.30	75.12	74.42
Removed in Layer-4	83.26	65.14	75.25	74.55
Removed in Layer-5	82.73	64.89	74.76	74.13

Table A. Ablation results on the effect of CMT in different layers.

Memory Bank	3D Object Detection (%)			
Sizes	Car	Ped.	Cyc.	mAP
256	83.25	66.15	76.03	75.14
512	83.32	66.14	76.12	75.19
1024	**83.58**	**66.45**	76.22	**75.42**
2048	83.26	66.21	**76.37**	75.28
4096	82.97	66.13	76.09	75.06

Table B. Ablation results by using various memory bank sizes in the OMDA module.

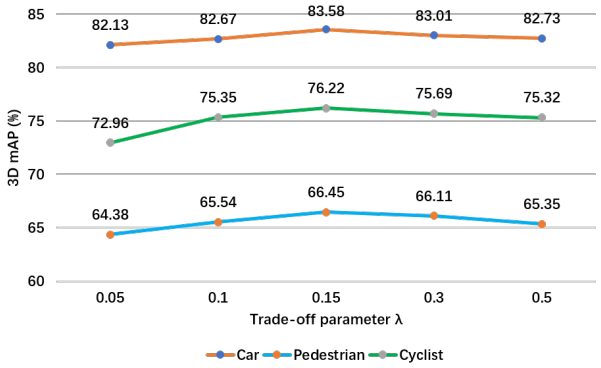

Figure B. Ablation results on the trade-off parameter λ, which balances the effects of the detection loss $(\mathcal{L}_{rpn}+\mathcal{L}_{rcnn})$ and the contrastive learning loss $(\mathcal{L}_{cl-p}+\mathcal{L}_{cl-o})$.

In regard of the **hyper-parameter λ**, as shown in Fig. B, the contrastive learning loss is not fully used for supervision when λ is small, thus yielding worse performance. In contrast, when λ becomes too large, the credit of the detection loss is improperly suppressed, also incurring poor results. The two kinds of losses, *i.e.* $(\mathcal{L}_{rpn}+\mathcal{L}_{rcnn})$ and $(\mathcal{L}_{cl-p}+\mathcal{L}_{cl-o})$, achieve their optimal balance when $\lambda = 0.15$, which is therefore used as a default in our work.

We also add results of applying **GT-Paste** only on point-clouds in Table C and show that inconsistent data augmentation tends to degrade the results (even worse than that without GT-Paste). Instead, OMDA well addresses this issue.

In order to further investigate the contributions of ITB we add more results by **removing ITB and local/global transformers**. As PTB cannot be directly removed, we replace it by PointNet++. The results summarized in Table D highlight their effectiveness.

Method	Car	Ped.	Cyc.	mAP(%)
w/o GT-Paste	81.73	63.30	70.63	71.89
with GT-Paste on 3D points	81.13	61.15	69.42	70.57
with OMDA	83.58	66.45	76.22	75.42

Table C. Ablation results on GT-Paste and contrastive learning on KITTI Val.

Method	Car	Ped.	Cyc.	mAP(%)
Full model	83.58	66.45	76.22	75.42
w/o ITB	81.22	63.20	70.18	71.53
Replacing PTB by PointNet++	82.69	64.93	74.61	74.08
w/o Global Transformer	82.83	65.21	74.88	74.31
w/o Local Transformer	83.19	65.94	75.65	74.93

Table D. Ablation results on ITB and PTB on KITTI Val.

Method	Modality	Params (M)	Time (ms)	mAP (%)
AVOD-FPN	L+I	38.07	100	56.84
F-PointNet	L+I	12.45	167	57.86
EPNet	L+I	16.23	178	–
VPFNet [3]	L+I	–	200	65.99
PointTransformer	L	6.06	250	61.66
M3DETR	L	19.66	256	64.65
CAT-Det (**Ours**)	L+I	23.21	314	67.05

Table E. Comparison of mAP, size of parameters and runtime on KITTI Test.

Setting	**Easy**	**Moderate**	**Hard**
Car (Detection)	95.97	94.71	92.07
Car (Orientation)	95.95	94.57	91.88
Car (3D Detection)	89.87	81.32	76.68
Car (Bird's Eye View)	92.59	90.07	85.82
Pedestrian (Detection)	67.15	56.75	53.44
Pedestrian (Orientation)	52.75	43.86	41.15
Pedestrian (3D Detection)	54.26	45.44	41.94
Pedestrian (Bird's Eye View)	57.13	48.78	45.56
Cyclist (Detection)	87.94	80.70	73.86
Cyclist (Orientation)	87.79	80.25	73.41
Cyclist (3D Detection)	83.68	68.81	61.45
Cyclist (Bird's Eye View)	85.35	72.51	65.55

Table F. Detailed results (%) on the official KITTI test leaderboard by using CAT-Det.

Finally, we report the **runtime and the size of parameters** with comparisons to major multi-modal methods and transformer based ones. As in Table E, our method reaches a good trade-off, with the highest accuracy and moderately increased model size and inference time. In the future, we will explore compression techniques to reduce the complexity of the transformer based methods for practical use.

C. More Visualization Results

As in Fig. 7 in the main paper, we demonstrate a few visualization results by performing 3D detection on KITTI val split via CAT-Det. In this section, we display more results.

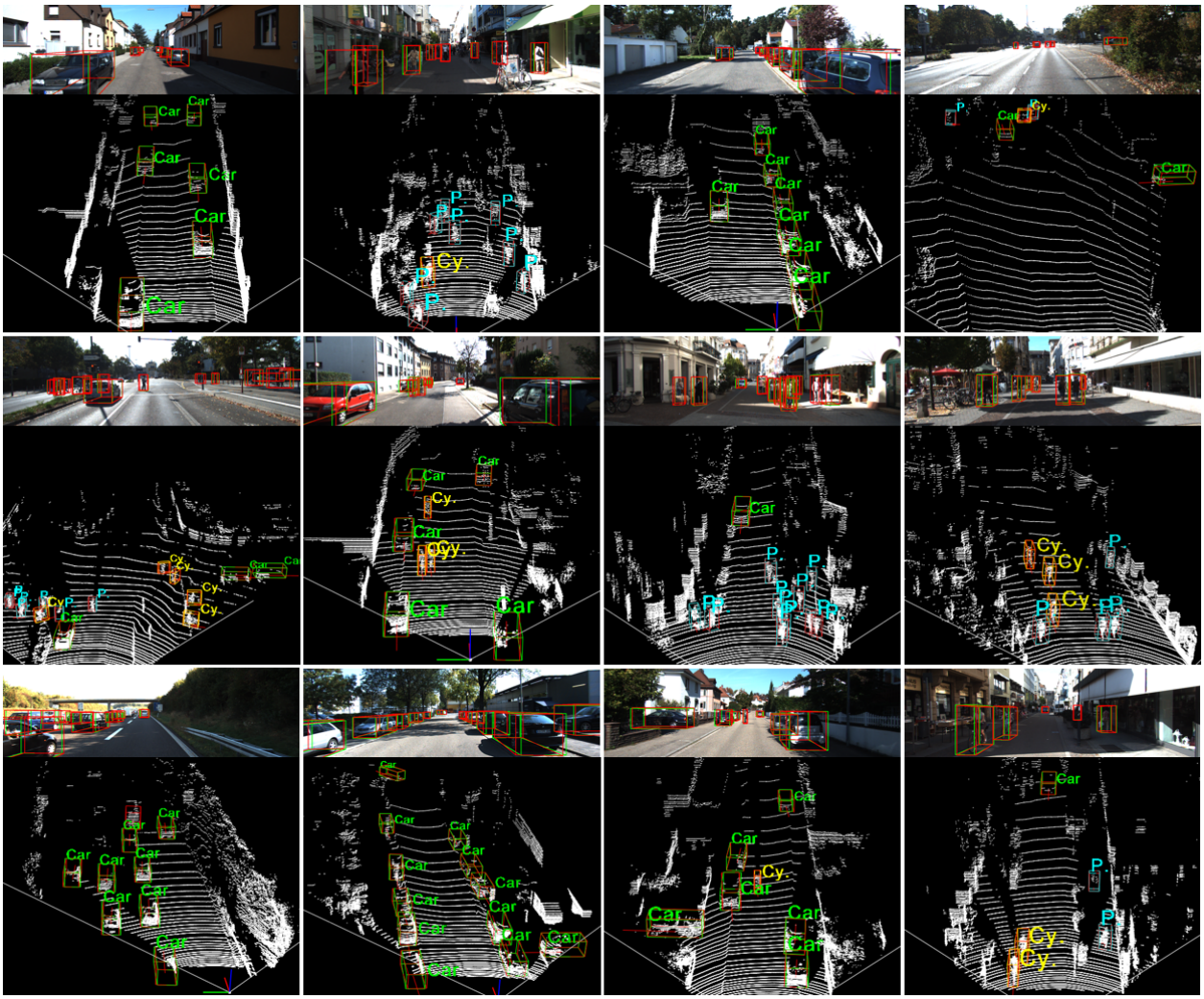

Figure C. Visualized results by CAT-Det on KITTI val split. Red/green rectangles indicate predicted/GT bounding boxes.

As shown in Fig. C, our approach precisely predicts both locations and orientations of 3D objects even under extremely challenging situations, including remote objects (the top row), tiny objects (the middle row) and objects with heavy occlusions (the bottom row).

D. Details on Official KITTI Test Leaderboard

In Table 1 and Table 2 from the main paper, we summarize the state-of-the-art results w.r.t AP/mAP on KITTI val/test splits. In this section, we provide more detailed results. Table F shows the official results in various settings (*i.e.* 3D, BEV, 2D and AOS) for three distinct levels of difficulties from the KITTI leaderboard. We also present the Precision-Recall curves in Fig. D on the test set.

References

[1] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017. 1

[2] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *CVPR*, pages 770–779, 2019. 1, 2

[3] Chia-Hung Wang, Hsueh-Wei Chen, and Li-Chen Fu. Vpfnet: Voxel-pixel fusion network for multi-class 3d object detection. *arXiv preprint arXiv:2111.00966*, 2021. 3

[4] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 1

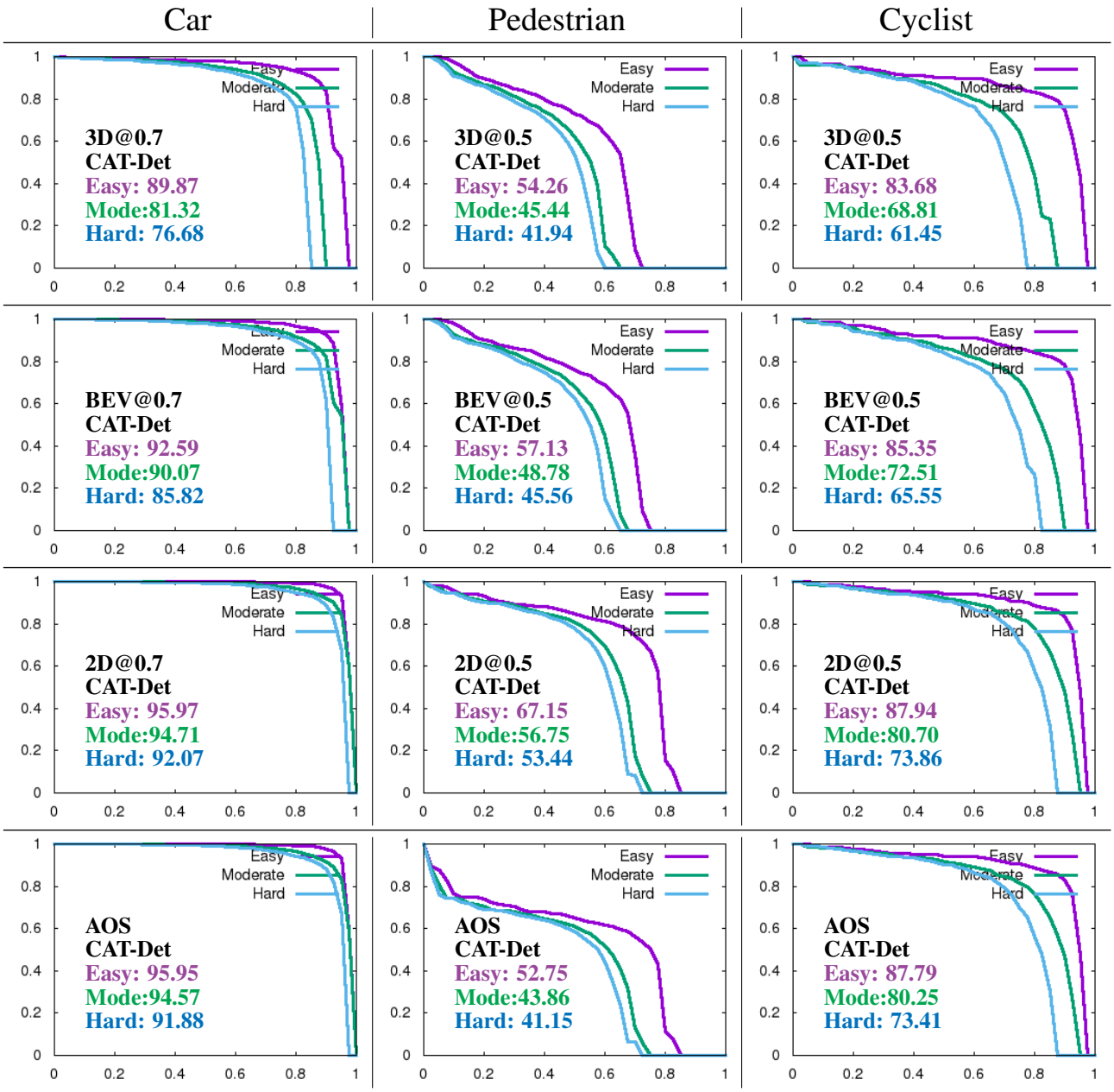

Figure D. **Precision Recall Curves** on the official KITTI test leaderboard by using CAT-Det. **Left to right:** Car@0.7, Pedestrian@0.5 and Cyclist@0.5. **Top to bottom:** 3D Detection, Bird's Eye View, 2D Detection and Orientation.