Learning Ego 3D Representation as Ray Tracing

Jiachen Lu[1], Zheyuan Zhou[1], Xiatian Zhu[2], Hang Xu[3], and Li Zhang[1⋆]

[1]Fudan University [2]University of Surrey [3]Huawei Noah's Ark Lab

https://fudan-zvg.github.io/Ego3RT

A Appendix

A.1 Projection matrix

In typical cases, we usually have one LIDAR coordinate (3D), N_{view} camera coordinate (3D) and N_{view} image coordinate (2D). First, a 3D point $\mathbf{x}_{\text{lidar}} = (x, y, z, 1)$ in the rectified LIDAR coordinate will be transformed to $\mathbf{x}_{\text{cam}}^{(t)} = (x', y', z', 1)$ in the t^{th} rectified camera coordinate with a given matrix $\mathbf{M}_{\text{ex}}^{(t)}$ called extrinsic parameter. Next, $\mathbf{x}_{\text{cam}}^{(t)} = (x', y', z', 1)$ is projected to a point $\mathbf{x}_{\text{img}}^{(t)} = (u, v, 1)^\top$ in the t^{th} image plane by

$$\mathbf{x}_{\text{img}}^{(t)} = \mathbf{M}_{\text{in}}^{(t)} \mathbf{x}_{\text{cam}}^{(t)}, \quad \mathbf{M}_{\text{in}}^{(t)} = \begin{pmatrix} f_u^{(t)} & 0 & c_u & -f_u^{(t)} b_x^{(t)} \\ 0 & f_v^{(t)} & c_v & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{1}$$

Here, $\mathbf{M}_{\text{in}}^{(t)}$ is the projection matrix for t^{th} camera. (f_u, f_v) is the focal length, (c_u, c_v) is the location of optical center and $b_x^{(t)}$ denotes the baseline with respect to reference camera (0 for nuScenes). In case of $0 < u, v < 1$, the point will be projected inside the image, otherwise outside.

A.2 Downstream task head.

Segmentation head. For the BEV segmentation task, we choose a group of progressive up-sampling convolution-based semantic segmentation decoder heads to deal with different elements from the map. Technically, a 1×1 Conv layer, a batch norm layer with ReLU, and a bilinear upsample Conv layer together form one up-sampling module. The decoder heads for predicting different map elements use the exact BEV features after the BEV encoder.

A.3 Objective functions

There are two training objectives for our model, including the loss $\mathcal{L}_{det}$ for object detection, and the loss $\mathcal{L}_{seg}$ for map elements segmentation:

$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{seg}. \tag{2}$$

⋆ Li Zhang (lizhangfd@fudan.edu.cn) is the corresponding author with School of Data Science, Fudan University.

Detection To handle the severe class imbalance with the nuScenes dataset, following CBGS [2] we group the similar classes into the same sub-task head. We use the focal loss for classification to alleviate the sample imbalance during our training, and simply adopt $L1$ loss to regress the normalized box parameters.

The classification loss for a specific sub-task $\mathcal{L}_{cls}^{t}$ is formulated as follows:

$$\mathcal{L}_{cls}^{t} = -\frac{1}{N}\sum_{y_{cls}\in Y_{cls}} \begin{cases} (1-\hat{y}_{cls})^{\alpha}\log(\hat{y}_{cls}) & \text{if } y_{cls}=1 \\ (1-y_{cls})^{\beta}(\hat{y}_{cls})^{\alpha}\log(1-\hat{y}_{cls}) & \text{otherwise} \end{cases}, \tag{3}$$

where Y_{cls} and N represents the set of pixels on the heatmap and the number of objects in t-th group, respectively. $\hat{y}_{cls}$ is the predicted classification probability and y_{cls} is the ground-truth. α and γ are the parameters of the focal loss [1].

The 3D bounding box regression loss for a specific sub-task $\mathcal{L}_{box}^{t}$ could be formulated as:

$$\mathcal{L}_{box}^{t} = \sum_{res\in\mathcal{R}} \mathcal{L}_{L1}(\widehat{\Delta_{res}}, \Delta_{res}), \tag{4}$$

where $\widehat{\Delta_{res}}$ is the predicted residual for the candidate center and Δ_{res} is the target ground-truth. $\mathcal{R}$ is the set of a box parameters, where x, y are the refinement for the location, z stands for the height, l, h, w are the 3D bounding box size, $\sin\theta$ and $\cos\theta$ are the rotation at yaw angel, v_x, v_y represent the velocities of the object.

Therefore, the overall detection loss $\mathcal{L}_{det}$ is formulated:

$$\mathcal{L}_{det} = \sum_{t\in\mathcal{T}_{det}} (\lambda_{cls}\mathcal{L}_{cls}^{t} + \lambda_{box}\mathcal{L}_{box}^{t}), \tag{5}$$

where $\mathcal{T}_{det}$ stands for the set of sub-task groups, λ_{cls} and λ_{box} represent the loss weights for classification and box regression.

Segmentation We use 5 different segmentation heads for the static elements in the BEV map, and the pixel-wise binary cross-entropy loss $\mathcal{L}_{seg}^{t}$ for t-th sub-task. The overall segmentation loss $\mathcal{L}_{seg}$ is computed as follows:

$$\mathcal{L}_{seg} = \sum_{t\in\mathcal{T}_{seg}} \lambda_{seg}^{t}\mathcal{L}_{seg}^{t}, \tag{6}$$

where $\mathcal{T}_{seg}$ represents the set of elements in the BEV map, λ_{seg}^{t} is the loss weight of the element.

A.4 Additional ablation studies

Image backbone. We first provide results with different image feature extractors in Table 1. It presents that the learning of 3D representation relies highly on 2D representation.

Table 1: Comparison of different image feature extractors. † means the image feature extractor is initialized from a FCOS3D checkpoint. ‡ means the image feature extractor is initialized from a DD3D checkpoint.

Backbone	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓	mAP↑	NDS↑
ResNet50	0.706	0.281	0.663	0.964	0.249	0.332	0.380
ResNet101	0.714	0.275	0.421	0.988	0.292	0.355	0.409
ResNet101 †	0.657	**0.268**	0.391	0.850	0.206	0.375	0.450
VoveNet‡	**0.582**	0.272	**0.316**	**0.683**	**0.202**	**0.478**	**0.534**

Table 2: Ablation on each component of Ego3RT. For the baseline, we set $N_{point} = 1$ (w/o "looking around"), eliminate adaptive attention mechanism (w/o "adaptive looking") and polarization (including both polarized grid and polar attention).

Components	mAP↑	NDS↑
baseline	0.353	0.427
$+N_{point} = 3$	0.360	0.433
+adaptive attention	0.365	0.437
+polarization	0.375	0.450

Which leads to improvement To further clarify, we summarize the effect of each component in Table 2, including the choice of N_{point}, adaptive attention mechanism, polarized grid and polar attention. Importantly, each component of our Ego3RT yields good gain.

A.5 Additional qualitative results

Visualization with video On our page, we simultaneously generate visualization of dynamic object detection and static semantic segmentation results from the 3D representation. In specific, we project all bounding boxes of class *vehicle* in nuScenes from the detection head onto the generated BEV segmentation map for a clear comparison.

Visualization of object detection results Figure 3 presents visualization of object detection results of two scenes in nuScenes `val` set. We have the following observations. (i) Ego3RT yields precise localization regarding to the bird's-eye-view visualization, even for the objects at long distance. (ii) Ego3RT can still work well in rainy whether shown in the second scene, proving its robustness to the whether condition. (iii) There are some miss-labeling in this dataset. For example, traffic cone in the `BACK_RIGHT` image of second scene is mis-labeled as barrier, but Ego3RT correctly labels it as traffic cone.

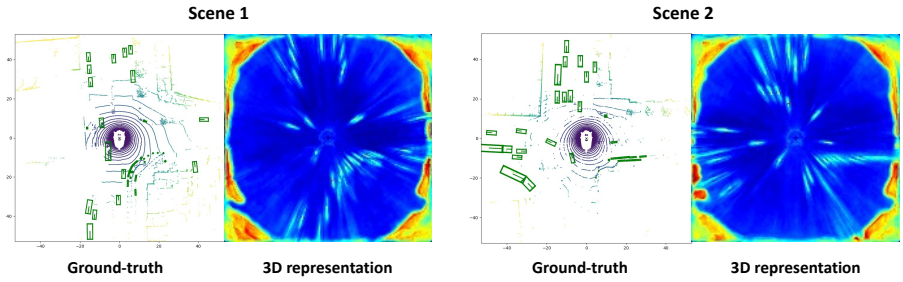
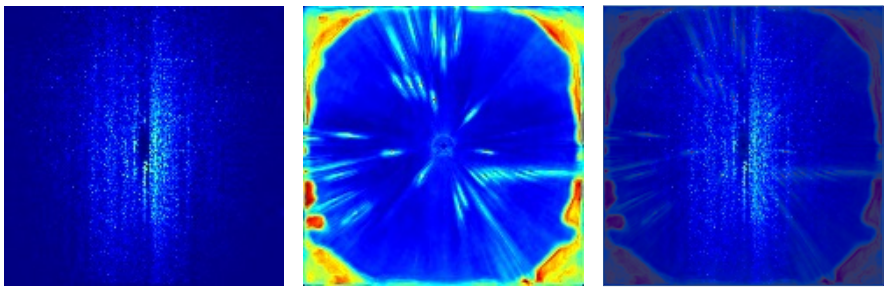


Fig. 1: Visualization results of 2 scenes' 3D representations given by Ego3RT. For each scene, *left* is the ground-truth of objects in bird's-eye view, while *right* is the visualization of 3D representation. Colors closer to `Red` represent higher response while colors closer to `Blue` represent lower response.

Visualization of 3D representation We provide the visualization of Ego3RT's learned 3D representation of the same scenes shown in the last section in Figure 1. The 3D representations predicted by Ego3RT are simply taken average on the channel to visualize. There is clear activation in the 3D representation wherever there is an object. The visualization demonstrates that Ego3RT actually learns 3D dense representation.

Objects' localization distribution There is an interesting observation that the outer part of 3D representation has different pattern in comparison with the inner part. At the beginning, we considered it was caused by the error in codes, but this different pattern remained even after a careful inspection. It is not until we visualized the objects' localization distribution of nuScenes that the answer was uncovered. Objects in nuScenes dataset appear more frequently at the center than the surrounding area. As is shown in Figure 2(c), the boundary of 3D representation's inner part well matches that of the objects' localization distribution. Therefore, Ego3RT reveals some data distribution while reasoning the 3D representation.



(a) (b) (c)

Fig. 2: (a) Distribution (heat map) of object localization in bird's-eye-view on nuScenes dataset. Colors closer to `Red` represent higher frequency while colors closer to `Blue` represent lower frequency. (b) 3D representation generated by Ego3RT. Colors closer to `Red` represent higher response while colors closer to `Blue` represent lower response. (c) Distribution of object localization with the 3D representation, in the same coordinate as bird's-eye-view.

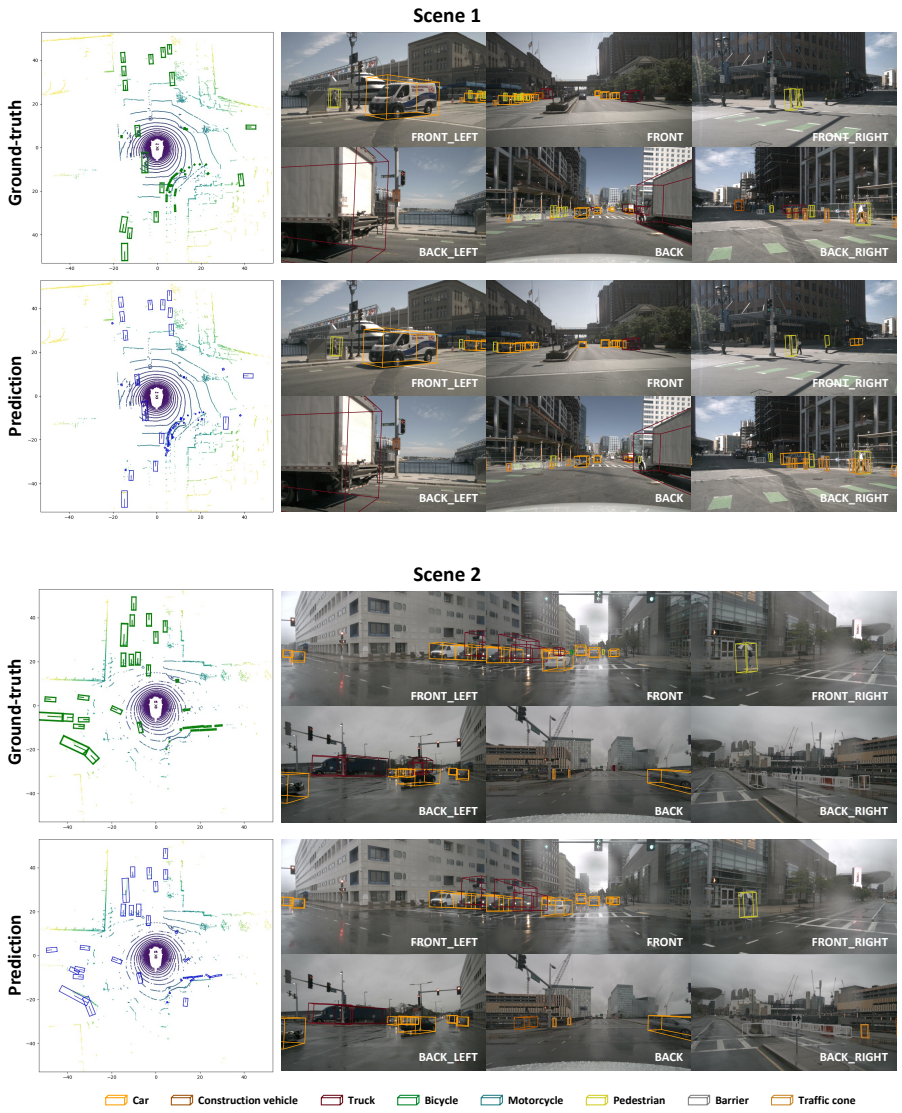


Fig. 3: Qualitative results on nuScenes dataset. Two scenes with both ground-truth and prediction are shown. *Left* are bird's-eye-view visualizations of object detection results. *Right* are in image perspective with prediction results. Different colors stand for different categories.

References

1. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV (2017) 2
2. Zhu, B., Jiang, Z., Zhou, X., Li, Z., Yu, G.: Class-balanced grouping and sampling for point cloud 3d object detection. arXiv preprint (2019) 2