Interactive 3D Object Detection with Prompts (Supplementary Material)

Ruifei Zhang[1,2⋆], Xiangru Lin[4⋆], Wei Zhang[4], Jincheng Lu[4], Xuekuan Wang[4], Xiao Tan[4], Yingying Li[4], Errui Ding[4], Jingdong Wang[4], and Guanbin Li[3,5⋆⋆]

[1] The Chinese University of Hong Kong, Shenzhen, China
[2] Shenzhen Research Institute of Big Data, Shenzhen, China
[3] Sun Yat-sen University, Guangzhou, China
[4] Department of Computer Vision Technology (VIS), Baidu Inc., China
[5] Peng Cheng Laboratory

1 Detailed Experiment Settings

1.1 Training Protocol

Throughout the training process, similar to common interactive algorithms [1, 9], we leverage Ground Truth to simulate practical user prompts and iterative refinements. Detailed training procedures for *Prompt in 2D, Detect in 3D* and *Detect in 3D, Refine in 3D* phases are provided as follows:

Prompt in 2D, Detect in 3D: Prompts, either as 2D boxes or points, are simulated by transposing the 3D Ground Truth into 2D BEV or camera perspectives, with random perturbations to mimic the actual user inputs. Specifically, we denote the bounding box projected from 3D Ground Truth as $G_b = (G_{x1}, G_{y1}, G_{x2}, G_{y2})$, with center position (G_{cx}, G_{cy}) and height G_h, width G_w. The simulated user prompts are calculated as follows:

$$S_{cx} = G_{cx} + G_w * \mathrm{U}(-\theta_w, \theta_w) \tag{1}$$

$$S_{cy} = G_{cy} + G_h * \mathrm{U}(-\theta_h, \theta_h) \tag{2}$$

$$S_w = G_w + G_w * \mathrm{U}(-\theta_w, \theta_w) \tag{3}$$

$$S_h = G_h + G_h * \mathrm{U}(-\theta_h, \theta_h) \tag{4}$$

$$S_b = (S_{cx} - \frac{S_w}{2}, S_{cy} - \frac{S_h}{2}, S_{cx} + \frac{S_w}{2}, S_{cy} + \frac{S_h}{2}) \tag{5}$$

$$S_p = (S_{cx}, S_{cy}) \tag{6}$$

⋆ Equal contribution.
⋆⋆ Corresponding author.

where S_b and S_p denote the constructed box prompt and point prompt respectively. U is uniform sampling. θ_w, θ_h determine the range of shift. We set $\theta_w, \theta_h = 0.5$ for boxes and 0.3 for points. Additionally, we introduce a filtering strategy for box prompts, eliminating distorted boxes with an $IoU(S_b, G_b)$ less than 0.5. The simulated box or point is randomly utilized as prompt input during our training, and some visual examples are presented in Fig. 1.

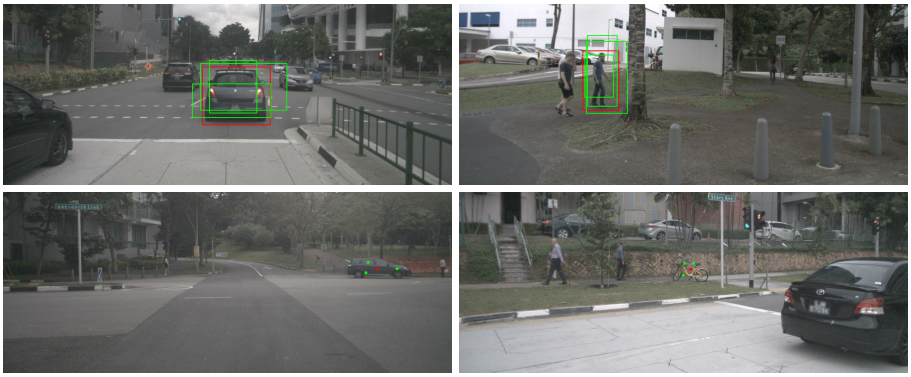

Fig. 1: Visual representation of 2D Prompts. For each object, the standard 2D bounding box G_b or point (G_{cx}, G_{cy}) projected from 3D Ground Truth is exhibited in red. Additionally, we present three randomly generated box or point prompts in green. Note that we only use one prompt in our experiments.

Detect in 3D, Refine in 3D: We decompose the 3D bounding box into five attributes, i.e. (gravity center, width, length, height and yaw angle), along with category information. To simulate the user interactive refinement, e.g. dragging the center point position or adjusting the orientation angle of the 3D bounding box, after obtaining the initial imperfect predictions based on the 2D prompts, we randomly replace the values of one (or any) attribute(s) with Ground Truth in our training process. The refined 3D prediction is encoded into the query sequences and expected to stimulate the model to rectify other characters in the next iteration.

1.2 Evaluation Protocol

To evaluate the effectiveness of our method, in addition to human annotation comparisons, we also conduct systematic and comprehensive simulated experiments across a wide range of real-world annotation scenarios and tasks:

– ***Annotation from Raw Data*** and ***Annotation from SOTA Model***. The former focuses on scenarios where annotators perform annotation from raw data without leveraging pretrained 3D detectors. Due to the absence of human prompts, we conduct simulated experiments by applying projected

2D bounding boxes or their centers from 3D ground truth, with random displacements, to derive prompts. On the other hand, the latter is more consistent with practical annotations. We utilize the initial predictions from SOTA 3D detection models to initiate our algorithm and then iteratively refine them for more accurate annotations.
- ***Closed-set Annotation*** and ***Open-set Annotation***. The former adheres to the traditional 3D object detection protocol of the nuScenes dataset, featuring ten closed-set classes. However, in practice, annotators are frequently tasked with labeling open-set objects. To assess this challenging yet crucial requirement, we extract eight novel categories from the nuScenes dataset that are not included in the training process, enabling us to conduct annotation experiments.

Furthermore, we conduct comprehensive experiments on various versions of our method, including different prompt formats and refinement strategies:

- ***Box Prompts*** and ***Point Prompts***. Our method supports flexible prompt formats. Unless otherwise specified, **Ours** refers to our method using 2D bounding boxes as prompts, whereas **Ours-point** leverages 2D points as cues.
- ***Refinement Strategies***. Our method enables users to easily adjust specific bounding box attributes by dragging the predicted results and also allows them to manually correct object category information. In our evaluations, we explore various orders of adjustments to bounding box attributes and assess our model both **without** and **with** corrected category priors, denoted as **Ours** and **Ours***, respectively. * indicates the pre-calibration of category data. The attributes from Ground Truth are utilized to mimic user refinements. Unless specified, we prioritize following the adjustment sequence of [gravity center, yaw angle, height, width, length]. The experimental results of other refinement orders are presented in the following section 3.

2 Additional Implementation Details

Model Structure: We initialize our encoder's weight from CMT [8] and keep it fixed throughout training. Image and point cloud backbones are provided by VoVNet [4] and VoxelNet [10], respectively. Moreover, we incorporate FPN [5] to amalgamate multi-scale features across both modalities. Our decoder consists of six decoder layers, which are trained from scratch. For global-to-local refinement, we employ RoI Align [2] to isolate local features of 20×20 dimensions. The critical layer Z that determines the transition from global to local is set to 3. We configure the input image dimensions to 1600×640 and voxelize the point cloud to 0.075m. The point cloud's region of interest spans from -54.0m to 54.0m on the X and Y axes, and from -5.0m to 3.0m on the Z axis.
Training configurations: Our model is trained on 8 A100 GPUs for a cumulative of 15 epochs using CBGS [11]. The AdamW [6] optimization algorithm drives our model's learning, initiated at a rate of 1.0×10^{-4} and governed by a

Table 1: Annotation performance on nuScenes `val` set. (Ri) denotes the i-th iteration refinement. "C" and "L" are camera and LiDAR modality respectively.

Method	Modality	NDS↑	mAP↑
Annotation from Raw Data			
Ours-point	CL	**73.0**	**72.6**
Ours-point (R1)	CL	83.5 (+10.5)	86.3 (+13.7)
Ours-point (R2)	CL	86.4 (+13.4)	86.3 (+13.7)
Ours-point (R3)	CL	87.4 (+14.4)	86.9 (+14.3)
Ours-point (R4)	CL	89.4 (+16.4)	89.0 (+16.4)
Ours-point (R5)	CL	90.5 (+17.5)	89.5 (+16.9)

Table 2: Ablation study of backbone. "It." denotes the iteration.

It.	Backbone		NDS↑	mAP↑
	Image	Lidar		
0	ResNet-50	VoxelNet	73.8	75.2
0	VOV-99	VoxelNet	**75.0**	**76.2**

cyclical learning rate policy [7]. We assign a batch size of 16. λ_{cls} and λ_{reg} are set to 2.0 and 0.25, respectively.

3 More Experiments

Results of Point Prompts: Table 1 illustrates the annotation performance on the nuScenes validation set driven by our 2D point prompts, termed as **Ours-point**. In the first *"prompt in 2D, detect in 3D"* stage, Ours-point reaches an NDS score of 73.0% and mAP of 72.6%. Moreover, benefiting from the subsequent iterative refinement strategy, Ours-point further improves the detection performance, with gains of 10.5%, 13.4%, 14.4%, 16.4%, 17.5% in NDS, respectively. Note that here we keep the same adjustment order with box prompts, i.e. [gravity center, yaw angle, height, width, and length], without employing manual category correction.

Ablation Study: We conduct additional ablation studies to delve into our method and provide more insights for other researchers. In line with the paper, we utilize **Ours**, which starts from box prompts, to perform ablation studies.

1. Ablation Study of Backbone: We assess the impact of the backbone in the first *"prompt in 2D, detect in 3D"* stage. As shown in Table 2, substituting the image backbone with ResNet-50 [3], our method achieves 73.8% in NDS and 75.2% in mAP, a slight decrease compared to the utilization of VoV-99 [4].

2. Ablation Study of Refinement Order: In this section, we explore the influence of different refinement orders during the *"detect in 3D, refine in 3D"* stage. Two additional adjustment orders [width, length, height, gravity center, yaw angle] and [yaw angle, gravity center, width, length, height] are introduced, and experimental results are detailed in Table 3. Observing the experimental

Table 3: Ablation study of refinement orders. w, l, h, g, θ denote width, length, height, gravity center and yaw angle, respectively.

Method	Attribute	NDS↑	mAP↑
Annotation from Raw Data			
Ours	-	75.0	76.2
Ours (R1)	w	77.1 (+2.1)	78.3 (+2.1)
Ours (R2)	l	78.1 (+3.1)	78.8 (+2.6)
Ours (R3)	h	78.9 (+3.9)	79.1 (+2.9)
Ours (R4)	g	87.3 (+12.3)	89.5 (+13.3)
Ours (R5)	θ	90.4 (+15.4)	89.5 (+13.3)
Ours (R1)	θ	78.3 (+3.3)	76.5 (+0.3)
Ours (R2)	g	87.3 (+12.3)	87.9 (+11.7)
Ours (R3)	w	88.4 (+13.4)	88.5 (+12.3)
Ours (R4)	l	89.5 (+14.5)	89.1 (+12.9)
Ours (R5)	h	90.4 (+15.4)	89.5 (+13.3)

results with different adjustment orders, we can deduce a common pattern: fine-tuning the gravity center can maximize the model's self-correction ability, leading to significant performance gains.

4 Visualizations

In this section, we provide additional visualization results of our method, covering both *"prompt in 2D, detect in 3D"* and *"detect in 3D, refine in 3D"* stages. As shown in Fig. 2, **Ours** represents the initial 3D prediction driven by 2D prompts. **Ours (R1)** and **Ours (R5)** depict the results after the first and fifth refinements, respectively. We adhere to the adjustment order [gravity center, yaw angle, height, width, length]. **Ours (R1)** simulates user exclusively refining the gravity center, whereas **Ours (R5)** adjusts all five attributes. The outcomes highlighted by red circles illustrate that, with a single round of manual adjustment focusing solely on the gravity center, our model automatically rectified other attributes such as yaw angles, thus yielding more accurate 3D results. This evidence showcases that our model possesses self-correction capabilities, effectively diminishing the necessity for user interactions and enhancing annotation efficiency.

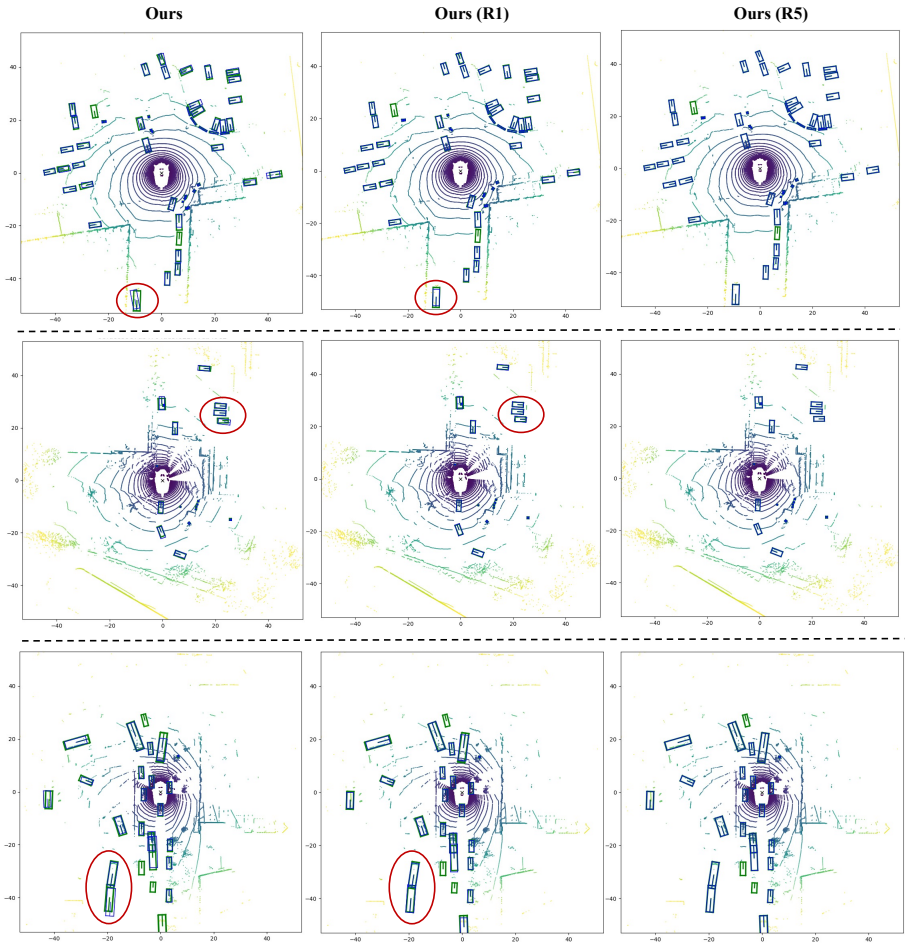

Fig. 2: Visualizations of the 3D detection results from our method are presented. Ground Truth and our predictions are exhibited in green and blue, respectively. **Ours** represents the initial 3D prediction driven by 2D prompts. **Ours (R1)** and **Ours (R5)** depict the results after the first and fifth refinements, respectively. It's noteworthy that we adhere to the adjustment order [gravity center, yaw angle, height, width, length]. **Ours (R1)** simulates user exclusively refining the gravity center, whereas **Ours (R5)** adjusts all five attributes. The outcomes highlighted by red circles illustrate that, with a single round of manual adjustment focusing solely on the gravity center, our model automatically rectified other attributes such as yaw angles, thus yielding more accurate 3D results. This evidence showcases that our model possesses self-correction capabilities, effectively diminishing the necessity for user interactions and enhancing annotation efficiency.

References

1. Chen, X., Zhao, Z., Zhang, Y., Duan, M., Qi, D., Zhao, H.: Focalclick: Towards practical interactive image segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1300–1309 (2022) 1
2. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2961–2969 (2017) 3
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016) 4
4. Lee, Y., Park, J.: Centermask: Real-time anchor-free instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13906–13915 (2020) 3, 4
5. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2117–2125 (2017) 3
6. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017) 3
7. Smith, L.N.: Cyclical learning rates for training neural networks. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 464–472. IEEE (2017) 4
8. Yan, J., Liu, Y., Sun, J., Jia, F., Li, S., Wang, T., Zhang, X.: Cross modal transformer via coordinates encoding for 3d object dectection. arXiv preprint arXiv:2301.01283 (2023) 3
9. Yang, J., Zeng, A., Li, F., Liu, S., Zhang, R., Zhang, L.: Neural interactive keypoint detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15122–15132 (2023) 1
10. Zhou, Y., Tuzel, O.: Voxelnet: End-to-end learning for point cloud based 3d object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4490–4499 (2018) 3
11. Zhu, B., Jiang, Z., Zhou, X., Li, Z., Yu, G.: Class-balanced grouping and sampling for point cloud 3d object detection. arXiv preprint arXiv:1908.09492 (2019) 3