# GSOT3D: Towards Generic 3D Single Object Tracking in the Wild

## Supplementary Material

In this supplementary material, we present more details and analysis as well as results of our work, as follows,

**S1 Mobile Robotic Platform**
In this section, we demonstrate more details of our mobile robotic platform used for multimodal data collection.

**S2 Annotation Tool**
We display more details of the annotation tool in labeling sequences with 9DoF 3D bounding boxes and its reliability analysis for high-quality annotation.

**S3 7DoF and 9DoF Parameterization**
We provide detailed explanations about 7DoF and 9DoF box parametrization.

**S4 More Statistics**
We demonstrate more statistics on GSOT3D regarding sequence length and per-category point density.

**S5 Analysis of Annotation Accuracy**
We analyze the accuracy of 3D annotations in GSOT3D.

**S6 Evaluation Metrics and 3D IoU**
We demonstrate detailed process on how to calculate the evaluation metrics and 3D IoU.

**S7 Formulation of Different 3D SOT Tasks**
We describe the formulation of different 3D SOT tasks.

**S8 Details of Feature Transformation Block**
We present the details of the feature transformation block adopted in our PROT3D.

**S9 Loss Function**
We present details of the loss function to train PROT3D.

**S10 Summary of Evaluated Trackers**
We offer a summary for trackers assessed on GOST3D.

**S11 Experiments on Unseen Categories**
We product experiments on unseen categories to evaluate the generalization capability of trackers.

**S12 Additional Discussions on GSOT3D**
We provide additional discussions on our GSOT3D.

**S13 Qualitative Results**
We offer qualitative analysis on GSOT3D.

**S14 Maintenance and Responsible Usage of GSOT3D for Research**
We discuss the maintenance and responsible usage of our proposed GSOT3D for research.

## S1 Mobile Robotic Platform

To collect multimodal data for GSOT3D, we build a mobile robotic platform based on Clearpath Husky A200. Multiple sensors, including a 64-beam LiDAR, an RGB camera and a depth camera, are deployed on the platform with careful calibration using the tool from [3]. Fig. 1 shows the picture of our mobile robotic platform for multimodal data acquisition in developing GSOT3D, and the specific configuration of sensors and robot chassis are listed in Tab. 1.

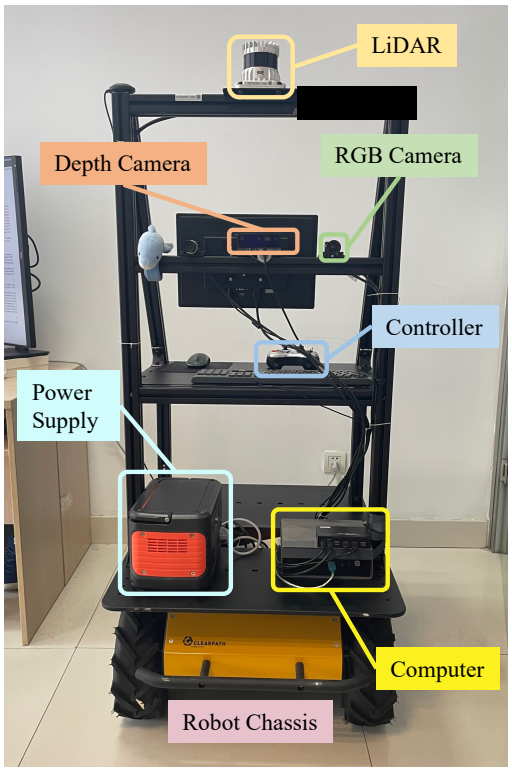



Figure 1. Our mobile robotic platform for data collection.

Table 1. Specific configuration of our mobile robotic platform.

| Device Name | Specification |
|---|---|
| LiDAR Sensor | Ouster OS-64 (64-beam) |
| Depth Camera | OAK D-Pro |
| RGB Camera | FLIR BFS-U3-32S4C-C |
| Robot Chassis | Clearpath Husky A200 |

## S2 Annotation Tool

For data labeling, we use the annotation tool provided by a company. The commercial software is to draw 3D boxes, and does not support auto-propagation of 3D box annotation. Fig. 2 shows the interface for 3D bounding box annotation. Specifically, for each point cloud frame, we perform initial annotation of the target object by drawing a 3D bounding box in the annotation region (note, this region can be flexibly

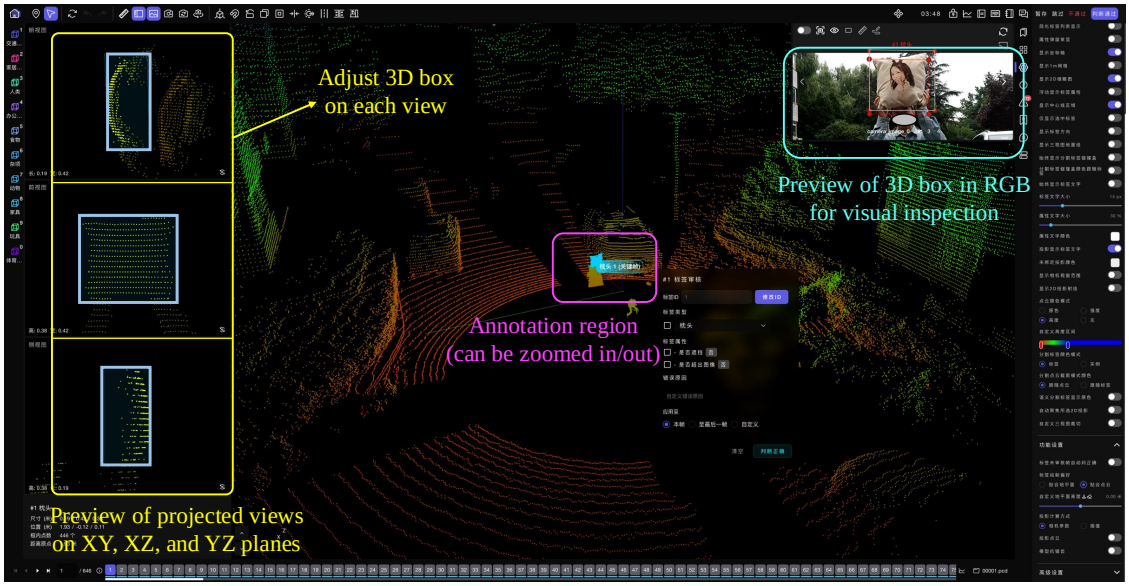
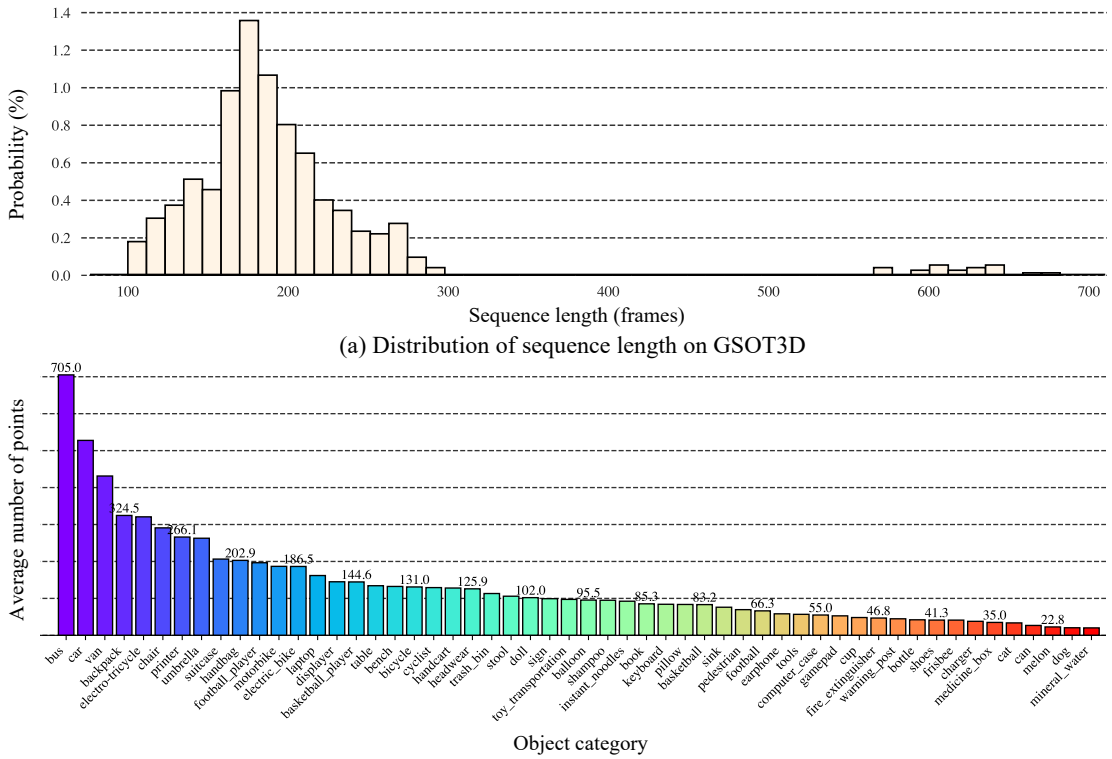



Figure 2. Annotation interface of our used annotation tool.




(a) Distribution of sequence length on GSOT3D

(b) Average number of points in each object category

Figure 3. Statistics on GSOT3D. Image (a): Distribution of sequence length. Image (b): Average number of points in each object category

zoomed in or out). Then, the initial 3D bounding box is refined by adjusting the 2D boxes on each projected view on XY, XZ, and YZ planes. In the annotation tool, a preview of the 3D box in the RGB image is provided for visual inspection of the refined box. By doing this, we can ensure the obtained annotation is reliable. Please note that, all the annotations from the labeler will be inspected careful by the experts (see this part in the main text) and further refined (by the same labeler) if necessary for high quality.

## S3 7DoF and 9DoF Parameterization

7DoF box has $(x, y, z, w, h, l, \theta_z)$, with $(x, y, z)$ the box center coordinates, $(w, h, l)$ the box width, height, and length, and $\theta_z$ the heading angle along $z$-axis, while 9DoF box has $(x, y, z, w, h, l, \theta_x, \theta_y, \theta_z)$ with the extra two $\theta_x$ and $\theta_y$ the heading angles along $x$- and $y$-axis. Compared to 7DoF considering rotation around one axis, 9DoF considers *full* rotations around all three axes and is more *flexible and accurate* to describe targets, thus having more application scenarios. For this reason, we adopt 9DoF box in our GSOT3D.

## S4 More Statistics

In this section, we demonstrate more statistics of GSOT3D. Specifically, Fig. 3 (a) shows distribution of sequence length on GSOT3D. Although the average length of GSOT3D is 198 frames, there exist several relatively longer ones with sequence length larger than 600 frames, which can be used for analyzing trackers on relatively longer sequences. Besides, Fig. 3 (b) demonstrates the average number of points for each category. We can clearly see that, the categories of *bus*, *car*, and *van* on average contain the most number of points, while the categories of *dog* and *mineral_water* consist of the least number of points. We hope this statistics can help readers better understand our GSOT3D.

## S5 Analysis of Annotation Accuracy

We follow Track-it-in-3D [16] to analyze the accuracy of 3D bounding box annotations in GSOT3D. In specific, we project 3D boxes onto 2D plane to obtain 2D annotations and then manually re-label 10% of data in GSOT3D with 2D boxes. We calculate the average projection error and average IoU between them, which is 8.9 pixels and 83%, indicating that our 3D annotation is reliable.

## S6 Evaluation Metrics and 3D IoU

Inspired by [8], we use Average Overlap (**AO**) and Success Rate (**SR**) as our indicators. The AO represents the average overlap between ground truth and estimated bounding boxes in a sequence, while the SR denotes the percentage of successful tracking frames with overlaps exceeding a threshold. The AO and SR of the $i^{\text{th}}$ sequence can be calculated via

$$\begin{aligned} \text{AO}_i &= \frac{1}{N_s}\sum_{j=1}^{N_s}\left[\Omega\left(p_j, g_j\right)\right] \\ \text{SR}_i &= \frac{1}{N_s}\sum_{j=1}^{N_s}\mathbb{I}\left(\Omega\left(p_j, g_j\right) \geq \tau\right) \end{aligned} \tag{1}$$

where $N_s$ represents the length of the $i^{\text{th}}$ sequence, $p_j$ and $g_j$ the predicted bounding box and ground truth (GT) in the $j^{\text{th}}$ frame. $\tau$ is the threshold for successful tracking. $\mathbb{I}\left(\cdot\right)$ is the Indicator Function, which takes the value 1 when the condition is met and 0 otherwise. $\Omega\left(p_j, g_j\right)$ denotes the intersection over union (IoU) of the GT and prediction for the $j^{\text{th}}$ frame, which can be written as follows,

$$\Omega\left(p_j, g_j\right) = \frac{p_j \cap g_j}{p_j \cup g_j}. \tag{2}$$

Unlike existing benchmarks (*e.g.*, KITTI [7], Track-it-in-3D [16]) that directly take the average on a frame-wise or sequence-wise, we leverage mean Average Overlap (**mAO**) and mean Success Rate (**mSR**) to measure different tracking algorithms on both sequence-wise and category-wise, similar to [8], aiming to provide class-balanced metrics that can reflect the general tracking performance. Specifically, mAO is calculated by averaging the class-wise overlaps, *i.e.*, 3D Intersection over Union (3D IoU, which will be described later), between all tracking results and the groundtruth, and mSR computes the class-wise percent of successful frames in which 3D IoU is larger than a threshold. mAO and mSR can be obtained as follows,

$$\begin{aligned} \text{mAO} &= \frac{1}{C}\sum_{c=1}^{C}\left(\frac{1}{|S_c|}\sum_{i\in S_c}\text{AO}_i\right) \\ \text{mSR} &= \frac{1}{C}\sum_{c=1}^{C}\left(\frac{1}{|S_c|}\sum_{i\in S_c}\text{SR}_i\right) \end{aligned} \tag{3}$$

where $C$ is the total number of object categories in GSOT3D, $S_c$ the set of all sequences belonging to category $c$. $\text{AO}_i$ represents AO for the $i^{\text{th}}$ sequence in $S_c$, and $\text{SR}_i$ denotes SR. $\text{mSR}_{50}$ and $\text{mSR}_{75}$ refers to mSR with thresholds of 0.5 and 0.75, respectively, when computing success rate.

**3D IoU.** Conventional 3D IoU often does not consider targets that have symmetric structure. However, in our GSOT3D, there exist many targets with symmetric structure, such as *ball*, *umbrella*, and so on (148 sequences in total involved with symmetric structure). In these cases, conventional 3D IoU cannot be used for accurate measurement by considering a fixed direction. To deal with this, we adopt the strategy employed in [1, 2] to calculate 3D IoU values between bounding boxes in arbitrary directions. Specifically, the predicted bounding box is rotated $k$ times along its axis of symmetry, and the prediction yielding the maximum 3D IoU among these $k$ rotations is selected as the final result. In our evaluation protocol, we set $k = 120$, as this configuration achieves efficient computation while maintaining negligible error margins in the final measurement. The detailed calculation process can be seen in [4].

Therefore, for non-symmetric targets, we adopt method as in KITTI [7] for 3D IoU calculation, while for symmetric targets, we use strategy as in [1, 2] for 3D IoU computation.

## S7 Formulation of Different 3D SOT Tasks

GSOT3D is a unique platform to broaden research direction in 3D SOT by supporting different tasks, comprising single-

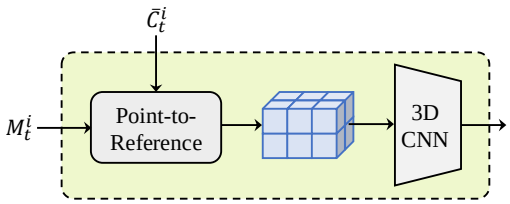



Figure 4. Architecture of the feature transformation block.

modal 3D object tracking, *i.e.*, *3D SOT on Point Cloud (PC)* (3D-SOT$_{\text{PC}}$), and multi-modal 3D tracking, *i.e.*, *3D SOT on RGB-PC* (3D-SOT$_{\text{RGB-PC}}$) or *RGB-Depth* (3D-SOT$_{\text{RGB-D}}$).

**3D-SOT$_{\text{PC}}$** aims at locating the target object on the point clouds. Given the PC sequence and the initial 9DoF 3D target box, the goal is to estimate a set of 3D bounding boxes to represent the target positions in the sequence. This process can be formulated as follows,

$$\{b_i\}_{i=2}^{N} \leftarrow \mathcal{T}_{\text{PC}}(\{\mathbf{p}_i\}_{i=1}^{N}, b_1) \tag{4}$$

where $b_i = (x_i, y_i, z_i, w_i, h_i, l_i, \alpha_i, \beta_i, \gamma_i)$ is the 9DoF 3D box in frame $i$ ($1 \leq i \leq N$), with $(x_i, y_i, z_i)$, $(w_i, h_i, l_i)$, and $(\alpha_i, \beta_i, \gamma_i)$ the target position, scale, and rotation angle. $b_1$ is given in the first frame and $\{b_i\}_{i=2}^{N}$ are predicted by the tracker $\mathcal{T}_{\text{PC}}$. $\{\mathbf{p}_i\}_{i=1}^{N}$ represent the PC sequence, and $N$ is the number of frames in the sequence.

Different from 3D-SOT$_{\text{PC}}$, **3D-SOT$_{\text{RGB-PC}}$** integrates the point clouds and RGB images for to locate target, aiming to improve 3D tracking using appearance information. It can be formulated as follows,

$$\{b_i\}_{i=2}^{N} \leftarrow \mathcal{T}_{\text{RGB-PC}}(\{\mathbf{p}_i\}_{i=1}^{N}, \{I_i\}_{i=1}^{N}, b_1) \tag{5}$$

where $b_1$ is the initial 9DoF 3D box, $\{b_i\}_{i=2}^{N}$ the predicted results by the tracker $\mathcal{T}_{\text{RGB-PC}}$, $\{\mathbf{p}_i\}_{i=1}^{N}$ and $\{I_i\}_{i=1}^{N}$ the PC and RGB image sequences, respectively.

Different than using PC, **3D-SOT$_{\text{RGB-D}}$** exploits a more economic way using RGB and depth images for 3D tracking, and can be formulated as follows,

$$\{b_i\}_{i=2}^{N} \leftarrow \mathcal{T}_{\text{RGB-D}}(\{D_i\}_{i=1}^{N}, \{I_i\}_{i=1}^{N}, b_1) \tag{6}$$

where $\mathcal{T}_{\text{RGB-D}}$ denotes the 3D tracker, $\{D_i\}_{i=1}^{N}$ are the depth image sequence, and all others are the same as in Eq. (5).

By supporting different tracking tasks, GSOT3D expects to expand research directions in 3D SOT.

## S8 Details of Feature Transformation Block

Fig. 4 displays the feature transformation block (FTB) used in each stage of our PROT3D. The feature transformation block is borrowed from [15] for its effectiveness. In specific, we first send the targetness mask $M_t^i$ and the point feature $\bar{C}_t^i$ to the Point-to-Reference operation, which is composed of a concatenation operation, a MLP, and an EdgeConv layer [13] for feature aggregation, as follows,

$$\begin{aligned} \hat{g}_t^i &= \text{Point-to-Reference}(\bar{C}_t^i, M_t^i) \\ &= \text{EdgeConv}(\text{MLP}(\text{Concatenate}(\bar{C}_t^i, M_t^i))) \end{aligned} \tag{7}$$

Table 2. Summary of evaluated trackers on GSOT3D.

| Tracker | Where | Backbone | Transformer |
|---|---|---|---|
| P2B [11] | CVPR'20 | PointNet++ | ✗ |
| BAT [17] | ICCV'21 | PointNet++ | ✗ |
| PTT [12] | IROS'21 | PointNet++ | ✓ |
| M2-Track [18] | CVPR'22 | PointNet | ✗ |
| CXTrack [14] | CVPR'23 | DGCNN | ✓ |
| MBPTrack [15] | ICCV'23 | DGCNN | ✓ |
| SeqTrack3D [9] | ICRA'24 | PointNet++ | ✓ |
| MS3SOT [10] | AAAI'24 | DGCNN | ✓ |

After this, the resulted feature $\hat{g}_t^i$ is fed into a 3D CNN network to generate point-wise feature. Fig. 4 illustrates FTB. For more details, please kindly refer to [15].

## S9 Loss Function

In this section, we present details regarding the loss function for training PROT3D. Specifically, after the $N^{\text{th}}$ stage, the final feature $\mathbf{x}_t^{N+1}$ is sent to the MLP layer for prediction. Similar to previous work [15], we use the following loss function for end-to-end training,

$$\mathcal{L}_{\text{total}} = \lambda_{\text{m}}\mathcal{L}_{\text{m}} + \lambda_{\text{c}}\mathcal{L}_{\text{c}} + \lambda_{\text{p}}\mathcal{L}_{\text{p}} + \lambda_{\text{s}}\mathcal{L}_{\text{s}} + \mathcal{L}_{\text{bbox}} \tag{8}$$

where $\mathcal{L}_{\text{total}}$ represents the total training loss, $\mathcal{L}_{\text{m}}$ the standard cross-entropy loss to supervise the targetness mask, $\mathcal{L}_{\text{c}}$ the mean square loss to supervise the target center, $\mathcal{L}_{\text{p}}$ the cross-entropy loss to supervise proposal score, $\mathcal{L}_{\text{s}}$ the cross-entropy loss to supervise the targetness score $\mathcal{S}_{\text{t}}$, and $\mathcal{L}_{\text{bbox}}$ the smooth-L$_1$ loss to supervise the 9DoF box $\mathcal{B}_t$ (including 3D center offset and 6D pose offset of size and angle). $\lambda_{\text{m}}$, $\lambda_{\text{c}}$, $\lambda_{\text{p}}$, $\lambda_{\text{s}}$ are hyper-parameters to balance different losses and are set to 0.2, 10.0, 1.0, and 1.0, respectively.

Our code will be publicly released, and more details can be found in our implementation.

## S10 Summary of Evaluated Trackers

To understand how existing trackers perform on GSOT3D and to provide comparison for future research, we assess eight representative trackers, including P2B [11], BAT [17], PTT [12], M2-Track [18], CXTrack [14], MBPTrack [15], SeqTrack3D [9], and M3SOT [10]. Please note that, these evaluated 3D trackers are point cloud-based, as almost all current 3D object trackers that share their implementations belong to this category. Tab. 2 summarizes these trackers.

## S11 Experiments on Unseen Categories

In order to further assess the generalization capability of our PROT3D on unseen categories, we conduct additional experiments with a different protocol. Specifically, in this new protocol, we use 40 classes for training and other unseen 14 classes for test. This enables assessing trackers on

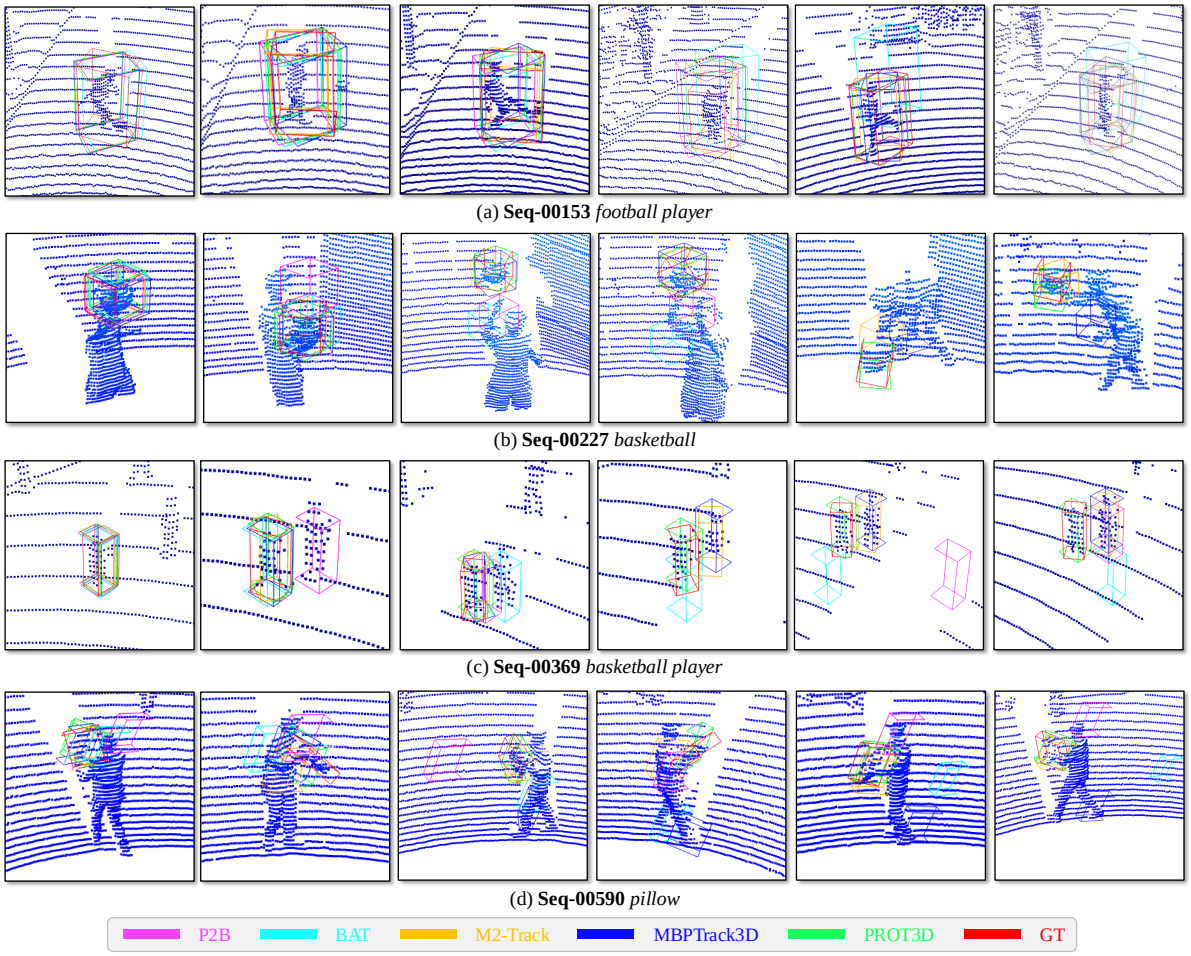



Figure 5. Qualitative results of several evaluated trackers and our proposed PROT3D. We can see that, the proposed PROT3D locates target object in different scenarios, showing its robustness for generic 3D object tracking.

Table 3. Experiments on unseen categories and comparison with performance under seen categories.

| Methods | Seen Test Categories | | | Unseen Test Categories | | |
|---|---|---|---|---|---|---|
| | mAO ↑ | $mSR_{50}$ ↑ | $mSR_{75}$ ↑ | mAO ↑ | $mSR_{50}$ ↑ | $mSR_{75}$ ↑ |
| M2-Track | 20.26 | 14.34 | 1.88 | 19.08 | 13.81 | 1.64 |
| MBPTrack | 20.54 | 16.55 | 2.57 | 19.76 | 14.93 | 2.24 |
| **PROT3D (ours)** | 21.97 | 19.76 | 5.22 | 21.03 | 18.65 | 4.82 |

unseen classes. The results of our method and comparison to other two state-of-the-art 3D point cloud trackers are in Tab. 3. As shown in Tab. 3, despite slight performance drop compared to the protocol with seen categories (in the main text), PROT3D shows promising results on unseen classes and surpasses other trackers, evidencing its effectiveness.

## S12 Additional Discussions on GSOT3D

**Discussion on Long-term Tracking.** Currently, GSOT3D is mainly focused on short-term 3D tracking, similar to [16]. Although its average sequence length is 198, there are sequences with more than 600 frames, which to some extent can reflect tracking performance in long-term scenarios. We are aware that in 2D object tracking (*e.g.*, LaSOT [5] and its extension [6]), the long-term 2D videos may consist of over thousands of frames. However, considering our current goal as well as the difficulties in manually collecting multi-modal sequences for 3D tracking, we leave the exploration of long-term 3D tracking dataset with average sequence length above a thousand and related algorithms to our future work.

**Discussion on the Extreme Scenario Conditions.** In our GSOT3D, we try our best to diversify scenarios when capturing data. Besides normal daytime scenarios, it includes some extreme scenarios conditions such as scenes with strong and weak light. Please note that, limited by our mobile platform

and related policies, it is hard for us to collect data in rainy and snowy weathers. This requires specific devices, and we will explore this in future.

## S13 Qualitative Results

In this section, we show qualitative results of different trackers and our PROT3D on GSOT3D in Fig. 5. From Fig. 5, we can see that, existing state-of-the-art trackers such as M2-Track, MBPTrack fail to accurately localize the target object in challenging scenarios with frequent occlusions and similar distractors, while our PROT3D can robustly locate the target in these cases owing to its progressive refinement strategy, showing its efficacy for generic 3D tracking.

## S14 Maintenance and Responsible Usage of GSOT3D for Research

**Maintenance.** Our GSOT3D will be hosted on the popular Github (all download links and our models will be publicly released). This enables conveniently checking the feedback from the community, and thus allows for improvements via necessary maintenance and updates by the authors. Besides, the authors will try their best to collect evaluation results of future trackers, aiming at providing up-to-date analysis and comparison on GSOT3D. Our ultimate goal is to develop a long-term and stable platform for 3D object tracking.

**Responsible Usage of GSOT3D.** GSOT3D aims to facilitate research and applications of 3D single object tracking. It is developed and used for ***research purpose only***.

## References


[1] Adel Ahmadyan, Liangkai Zhang, Artsiom Ablavatski, Jianing Wei, and Matthias Grundmann. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. In *CVPR*, 2021. 3

[2] Garrick Brazil, Abhinav Kumar, Julian Straub, Nikhila Ravi, Justin Johnson, and Georgia Gkioxari. Omni3d: A large benchmark and model for 3d object detection in the wild. In *CVPR*, 2023. 3

[3] Ankit Dhall, Kunal Chelani, Vishnu Radhakrishnan, and K Madhava Krishna. Lidar-camera calibration using 3d-3d point correspondences. *arXiv*, 2017. 1

[4] Christer Ericson. *Real-time collision detection*. Crc Press, 2004. 3

[5] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. Lasot: A high-quality benchmark for large-scale single object tracking. In *CVPR*, 2019. 5

[6] Heng Fan, Hexin Bai, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Harshit, Mingzhen Huang, Juehuan Liu, et al. Lasot: A high-quality large-scale single object tracking benchmark. *International Journal of Computer Vision*, 129:439–461, 2021. 5

[7] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 3

[8] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(5):1562–1577, 2021. 3

[9] Yu Lin, Zhiheng Li, Yubo Cui, and Zheng Fang. Seqtrack3d: Exploring sequence information for robust 3d point cloud tracking. In *ICRA*, 2024. 4

[10] Jiaming Liu, Yue Wu, Maoguo Gong, Qiguang Miao, Wenping Ma, Cai Xu, and Can Qin. M3sot: Multi-frame, multi-field, multi-space 3d single object tracking. In *AAAI*, 2024. 4

[11] Haozhe Qi, Chen Feng, Zhiguo Cao, Feng Zhao, and Yang Xiao. P2b: Point-to-box network for 3d object tracking in point clouds. In *CVPR*, 2020. 4

[12] Jiayao Shan, Sifan Zhou, Zheng Fang, and Yubo Cui. Ptt: Point-track-transformer module for 3d single object tracking in point clouds. In *IROS*, 2021. 4

[13] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics*, 38(5):1–12, 2019. 4

[14] Tian-Xing Xu, Yuan-Chen Guo, Yu-Kun Lai, and Song-Hai Zhang. Cxtrack: Improving 3d point cloud tracking with contextual information. In *CVPR*, 2023. 4

[15] Tian-Xing Xu, Yuan-Chen Guo, Yu-Kun Lai, and Song-Hai Zhang. Mbptrack: Improving 3d point cloud tracking with memory networks and box priors. In *ICCV*, 2023. 4

[16] Jinyu Yang, Zhongqun Zhang, Zhe Li, Hyung Jin Chang, Aleš Leonardis, and Feng Zheng. Towards generic 3d tracking in rgbd videos: Benchmark and baseline. In *ECCV*, 2022. 3, 5

[17] Chaoda Zheng, Xu Yan, Jiantao Gao, Weibing Zhao, Wei Zhang, Zhen Li, and Shuguang Cui. Box-aware feature enhancement for single object tracking on point clouds. In *ICCV*, 2021. 4

[18] Chaoda Zheng, Xu Yan, Haiming Zhang, Baoyuan Wang, Shenghui Cheng, Shuguang Cui, and Zhen Li. Beyond 3d siamese tracking: A motion-centric paradigm for 3d single object tracking in point clouds. In *CVPR*, 2022. 4