

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

Yifan Jiao^{1,2} Yunhao Li^{1,2} Junhua Ding³ Qing Yang³ Song Fu³ Heng Fan^{3†} Libo Zhang^{1†}

¹University of Chinese Academy of Sciences

²Institute of Software Chinese Academy of Sciences ³University of North Texas

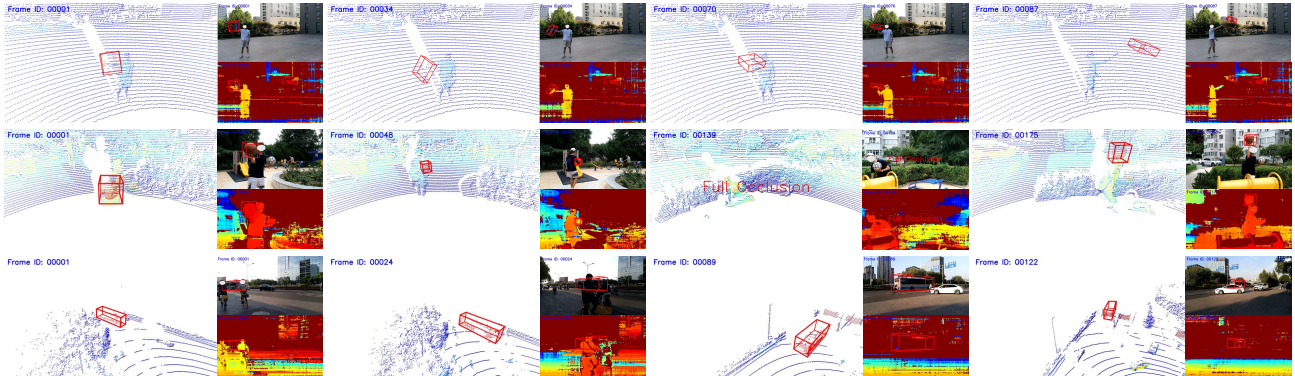


Figure 1. Demonstration of a few sequence samples from our GSOT3D. Each sequence is offered with multiple modalities, including *point cloud*, *RGB image*, and *depth*, supporting different 3D SOT tasks. *Best viewed in color and by zooming in for all figures in the paper.*

Abstract

In this paper, we present a novel benchmark, GSOT3D, that aims at facilitating development of generic 3D single object tracking (SOT) in the wild. Specifically, GSOT3D offers 620 sequences with 123K frames, and covers a wide selection of 54 object categories. Each sequence is offered with multiple modalities, including the point cloud (PC), RGB image, and depth. This allows GSOT3D to support various 3D tracking tasks, such as single-modal 3D SOT on PC and multi-modal 3D SOT on RGB-PC or RGB-D, and thus greatly broadens research directions for 3D object tracking. To provide high-quality per-frame 3D annotations, all sequences are labeled manually with multiple rounds of meticulous inspection and refinement. To our best knowledge, GSOT3D is the largest benchmark dedicated to various generic 3D object tracking tasks. To understand how existing 3D trackers perform and to provide comparisons for future research on GSOT3D, we assess eight representative point cloud-based tracking models. Our evaluation results exhibit that these models heavily degrade on GSOT3D, and more efforts are required for robust and generic 3D object tracking. Besides, to encourage future research, we present a simple yet effective generic 3D tracker, named PROT3D, that localizes the target object via a progressive spatial-temporal network and outperforms all

[†]Equal advising and co-last authors.

current solutions by a large margin. By releasing GSOT3D, we expect to advance further 3D tracking in future research and applications. Our benchmark and model as well as the evaluation results will be publicly released at our webpage <https://github.com/ailovejinx/GSOT3D>.

1. Introduction

As one of the most crucial problems in 3D computer vision, 3D single object tracking (SOT) aims to localize the desired target with a sequence of 3D bounding boxes, given its state in the first frame. Due to its key roles in many applications, such as intelligent vehicles, mobile robotics, navigation, etc, 3D object tracking has gained extensive attention in the past decade with many models proposed (e.g., [2, 3, 12, 28, 39]).

Current research mainly focuses on the point cloud (PC)-based 3D tracking. Relying on popular autonomous driving benchmarks (e.g., KITTI [11] and NuScenes [5]), numerous deep 3D trackers have been proposed and demonstrated state-of-the-art results (e.g., [25, 34, 36, 37]). Despite such progress, further development of *generic* 3D SOT is heavily *restricted* by currently adopted benchmarks due to several reasons: **(1) limited object classes**. To achieve general tracking capacity, a 3D tracker is expected to learn with sequences from a large set of categories during training. However, existing datasets for 3D SOT (e.g., [5, 11]), specially

designed for autonomous driving, comprise *very few* available categories (*e.g.*, 8 in [11] and 23 in [5]) for tracking, making them *inadequate* for designing generic 3D trackers. **(2) constrained scenarios.** In applications, a general tracker should be able to localize the target object under various scenarios, which requires it to be trained and assessed with sequences collected from diverse environments. Yet current datasets, due to their own specific aims, only offer sequences from the traffic scenario and thus are *unsuitable* for general tracking. **(3) restricted degrees of freedom (DoF).** For generic 3D tracking, a tracker needs to handle objects with arbitrary pose and size, often described with 9DoF consisting of 6D pose and 3D size. Nonetheless, currently used datasets [5, 11] comprise only targets of 7DoF, including 4D pose and 3D size, and thus are *undesirable* for developing general trackers locating arbitrary-pose objects.

It is worth noting that, besides the PC-based 3D SOT, the above autonomous driving datasets (*e.g.*, [5, 11]) can also be used for developing multi-modal, *i.e.*, RGB-PC, tracking by integrating point clouds and RGB images. Nevertheless, the aforementioned issues still exist, and therefore, limit the further development of generic 3D object tracking.

In addition to PC-based single- or multi-modal solutions, another direction that is more affordable is to leverage RGB and depth information for 3D tracking. For such a goal, a recent dataset [38] has been introduced by collecting RGB-D sequences from diverse categories and annotating each one with 9DoF 3D boxes. However, it is *limited* by its relatively *small scale*. In order to effectively train and reliably assess deep 3D trackers, it is desirable to have plenty of sequences in a dataset. Nonetheless in [38], there is a total of only 300 sequences with 36K frames, which might be *insufficient* for large-scale learning and evaluation of deep 3D trackers.

Contributions. To alleviate limitations in existing 3D SOT benchmarks and offer a versatile platform for 3D tracking, we introduce a high-quality benchmark, **GSOT3D**, which is dedicated to diverse generic 3D object tracking.

Specifically, our GSOT3D consists of 620 sequences and provides more than 123K frames in total. In order to ensure the diversity of GSOT3D, these sequences are carefully collected from a wide selection of 54 object classes from various environments. For each sequence in GSOT3D, multiple modalities, including the *point cloud (PC)*, *RGB image*, and *depth*, are offered using different sensors (see examples in Fig. 1). This allows GSOT3D to support different 3D tracking tasks, comprising the *single-modal* 3D SOT on PC and *multi-modal* 3D SOT on RGB-PC or RGB-D, and therefore broadens the research directions in 3D tracking. For precise dense annotations, all the sequences in GSOT3D are manually labeled using 9DoF 3D bounding boxes with multiple rounds of inspection and refinement. To our best knowledge, GSOT3D is, to date, the *largest* benchmark dedicated to generic 3D object tracking. Besides, it is the *first* bench-

mark, to date, that simultaneously supports different single- and multi-modal 3D SOT tasks.

Compared with existing benchmarks (*e.g.*, [5, 11]) with a few object classes for 3D SOT on PC and RGB-PC in traffic scene, GSOT3D is more *diverse* by containing 54 categories and various scenarios, making it more favorable for generic 3D tracking. Moreover, compared to [38] consisting of 300 sequences with 36K frames for RGB-D 3D tracking, GSOT3D is *larger* by providing 620 sequences ($2\times$ larger) with 123K frames ($3\times$ larger), and hence more desirable for large-scale learning and evaluation of deep 3D tracking.

In order to understand how existing 3D trackers perform and to provide comparisons for future research, we assess 8 representative PC-based tracking methods. Please note that, compared to 2D generic object tracking, there are *not* many open-sourced 3D trackers and most methods are PC-based. For this reason, we finally include 8 PC-based trackers, that are representative and provide executable implementations, for evaluation. Our evaluation reveals that, not surprisingly, all current models degrade severely on the more challenging GSOT3D, which demonstrates the difficulty in achieving generic 3D tracking in the real-world, and more efforts are needed for future improvements.

Moreover, to facilitate research on GSOT3D, we present a simple but effective generic 3D tracker, dubbed **PROT3D**, for *class-agnostic* 3D tracking on point clouds. The core of PROT3D is a progressive spatial-temporal architecture containing multiple stages. In each stage, target localization is performed by spatial-temporal matching with Transformer, and the result is applied to refine search region feature. The refined search region feature from one stage is forwarded to next stage for further improvements, and tracking result is generated after the final stage. This way, PROT3D gradually learns more discriminative features via progressive feature refinement, making it capable of handling more complex scenarios for generic tracking. It is worth noticing, unlike current trackers predicting a 7DoF box, our PROT3D produces a 9DoF box for more precise tracking. Despite its simplicity, PROT3D outperforms all other methods, and expects to provide a reference for future research.

In summary, our contributions are as follows: ♠ We propose a new benchmark GSOT3D comprising 620 sequences with more than 123K frames to facilitate 3D object tracking; ♥ GSOT3D provides multiple modalities to each sequence, making it a versatile platform for various research directions in 3D tracking; ♣ We evaluate eight representative trackers to understand their performance and to offer comparisons to future research; ♦ We present a simple yet effective tracker, PROT3D, to encourage future research on GSOT3D.

2. Related Work

Benchmarks for 3D Single Object Tracking. Datasets are crucial for 3D single object tracking by providing platforms

Table 1. Detailed comparison of our GSOT3D with existing 3D SOT benchmarks. O: Outdoor, I: Indoor, PC: Point cloud, D: Depth. Please notice that, we gray KITTI and NuScenes, as they are *not* specifically developed for 3D single object tracking. ¶: Based on the information provided in the original paper [38], there are 44 object categories in total in Track-it-in-3D.

Benchmark	Where	Total Sequences	Total Frames	Avg. Length	Object Classes	Data Scenarios	Modality			3D SOT Task on		
							RGB	PC	Depth	PC	RGB-PC	RGB-D
KITTI [11]	CVPR'2012	21	15K	-	8	O	✓	✓	✗	✓	✓	✗
NuScenes [5]	CVPR'2020	1,000	40K	-	23	O	✓	✓	✗	✓	✓	✗
Track-it-in-3D [38]	ECCV'2022	300	36K	120	44 [¶]	I & O	✓	✗	✓	✗	✗	✓
GSOT3D (ours)	-	620	123K	198	54	I & O	✓	✓	✓	✓	✓	✓

for training and assessment. Currently, the popular datasets, particularly for 3D tracking on point cloud, are mainly borrowed from the autonomous driving benchmarks, including KITTI [11] and NuScenes [5]. Specifically, KITTI comprises 21 sequences with 15K frames, and each one is offered with point clouds and RGB images. Similar to KITTI but with a larger size, NuScenes comprises 1,000 sequences with 40K frames. Since KITTI and NuScenes are originally designed for autonomous driving, they usually need appropriate conversions before being used for 3D SOT. Besides KITTI and NuScenes for point cloud-related 3D SOT, the work of [38] recently proposes a new benchmark, named Track-it-in-3D, dedicated to RGB-D-based 3D object tracking. It contains 300 sequences with 36K frames, collected from 44 classes. Each sequence is annotated with 9DoF 3D boxes for more precise generic 3D object tracking.

Despite the above benchmarks, the further development of 3D SOT remains constrained by the limitations discussed earlier, which motivates our GSOT3D in this work, a versatile dataset dedicated to different generic 3D tracking tasks. Tab. 1 compares our GSOT3D with other datasets in detail.

3D Object Tracking Algorithms. 3D tracking has received extensive attention in the past decade. Most recent research focuses on point cloud-based 3D object tracking. The seminal work of [12] adopts a Siamese network that explores the shape completion for 3D tracking on point clouds. In order to improve the efficiency and enhance the performance, the work of [28] introduces an end-to-end framework that integrates target proposal and verification for 3D tracking. The method of [39] leverages prior information from the target box to enhance features for improvement. The work of [40] explores the motion cues from a sequence for 3D tracking, displaying promising results. The method of [15] proposes to improve tracking performance on sparse point clouds by learning shape-aware features and localizing the target from the dense bird’s eye view (BEV) feature maps, boosting the tracking results. More recently, inspired by [30], the Transformer has been extensively used for 3D tracking, showing excellent results [13, 16, 23, 25, 29, 33, 34, 36, 37, 41].

Besides 3D tracking on point clouds, another direction is to leverage RGB and depth information for 3D SOT. The work of [3] introduces a part-based 3D tracker using sparse learning. In [38], a Siamese network is proposed to fuse the

RGB and depth information for RGB-D 3D tracking.

Generic 2D Tracking Datasets. Our GSOT3D in this work is inspired, to some extent, by existing generic 2D tracking datasets. Early datasets, such as [10, 18–20, 24, 35], mainly aim at evaluating and comparing the tracking performance, and are usually small-scale. Later, to facilitate development of generic tracking in deep learning era, several large-scale tracking datasets (*e.g.*, [8, 14, 24, 27, 31]) have been developed by offering abundant videos. Particularly, these large benchmarks often include a diverse selection of categories, well enhancing the generalization ability of deep trackers.

Sharing a similar goal with current large-scale 2D tracking benchmarks, GSOT3D aims at providing sufficient sequences from rich classes for generic 3D tracking. It is worthy to note that, compared to current large-scale 2D tracking benchmarks (*e.g.*, [8, 14, 24, 27, 31]) with over a thousand or tens of thousands videos, GSOT3D is relatively smaller due to the extreme difficulty in collecting sequences and annotating them using the 9DoF bounding boxes. That being said, GSOT3D to date is still the largest dataset that is dedicated to generic 3D single object tracking.

3. The Proposed GSOT3D Benchmark

3.1. Construction Principle

GSOT3D aims at serving as a *versatile* platform to facilitate different 3D tracking tasks with sufficient sequences and rich classes as well as high-quality annotations. To this end, we follow several principles when constructing GSOT3D:

- *Rich Object Class.* To achieve generic tracking, it is desirable to encompass diverse object categories in both training and evaluation. For this purpose, the new benchmark is expected to cover at least 50 categories, including common targets suitable for 3D tracking in our daily life.
- *Different 3D Tracking Tasks.* To broaden research directions in 3D SOT, multiple modalities should be provided for the sequences, allowing researchers to flexibly explore various 3D tracking tasks using different input types (single or multiple modalities) based on their specific needs.
- *Appropriate Scale.* To effectively train and evaluate deep trackers, sufficient sequences are needed for a benchmark. Considering the difficulty in collecting and labeling data for 3D tracking, we hope to gather at least 600 sequences

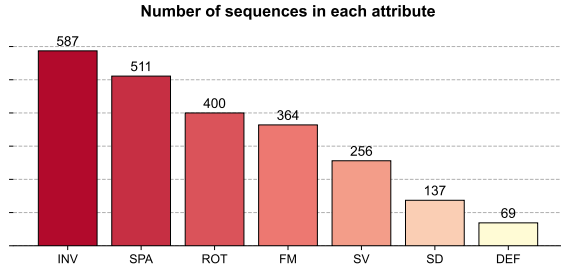


Figure 3. Distribution of videos per attribute.

Table 2. Comparison of training and test sets of GSOT3D.

	Total Sequences	Total Frames	Ave. Frames	Object Classes
GSOT3D _{Tra}	435	83,950	193	54
GSOT3D _{Tst}	185	39,740	215	54

which is assigned when the target is partially or fully invisible due to occlusion and/or out of view, *deformation* (DEF), which is assigned when the target is deformable, *fast motion* (FM), which is assigned when target moves larger than half size of its bounding box, *rotation* (ROT), which is assigned when target rotates in the view, *scale variation* (SV), which is assigned when the ratio of the 3D box is beyond [0.75, 1.5], *Similar Distractors* (SD), which is assigned when there exist similar targets in the view, and *Sparsity* (SPA), which is assigned when target information (point cloud or appearance) is sparse, *i.e.*, the target region contains less than 50 points on PC or 1,000 pixels on RGB or depth. For each sequence, a 7D binary vector is used to indicate the presence of an attribute: “1” for presence, and “0” otherwise.

Fig. 3 demonstrates the distribution of attributes. We can see that the most common attribute is INV, which may cause severe feature degradation for tracking. Besides, SPA and ROT frequently happen in sequences. We also notice, there are a few sequences involved with DEF, as some targets belonging to the human and animal meta classes are non-rigid, making the localization of them more challenging.

3.5. Dataset Split, Evaluation Protocol, and Tasks

Dataset Split. Our GSOT3D includes 620 multi-modal sequences, and we adopt the 70/30 principle to generate training and test splits. In specific, 435 sequences are utilized in the training set named GSOT3D_{Tra}, and the rest 185 for test set dubbed GSOT3D_{Tst}. Both GSOT3D_{Tra} and GSOT3D_{Tst} contain all the 54 object categories. In the dataset split, we try our best to make the distributions of these two sets close to each other. Tab. 2 displays the comparison of GSOT3D_{Tra} and GSOT3D_{Tst}, and the detailed splits will be released on our project paper together with our data and other materials.

Evaluation Protocol. Inspired by [14], we leverage mean Average Overlap (mAO) and mean Success Rate (mSR) for

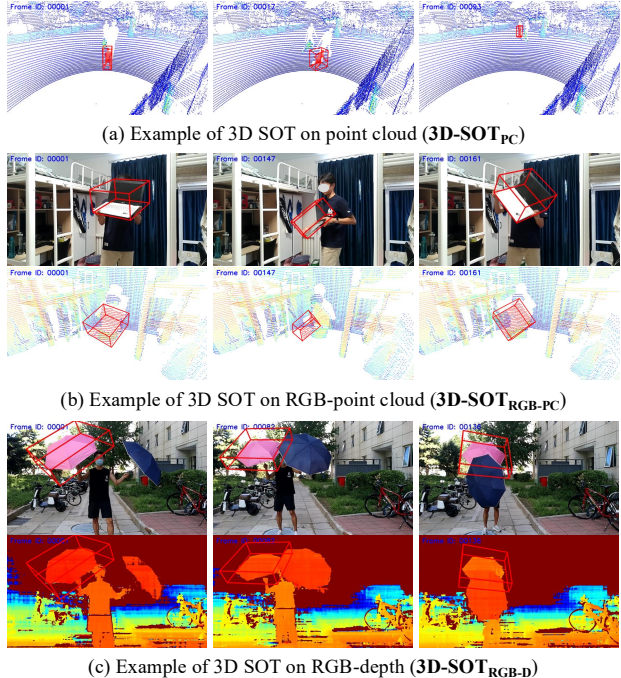


Figure 4. Illustration of different 3D SOT tasks on GSOT3D.

evaluation. mAO is computed by averaging the class-wise overlaps, *i.e.*, 3D Intersection over Union (or 3D IoU), between all tracking results and the groundtruth, while mSR measures class-wise percent of successful frames in which 3D IoU is larger than a threshold (*e.g.*, 0.5 or 0.75). The details of how to compute mAO and mSR as well as 3D IoU for different cases (non-symmetric and symmetric objects) can be seen in the **supplementary material**.

Please notice here, we do *not* utilize the precision metric as in previous studies for evaluation, because the precision, that measures the center points between tracking results and groundtruth, *cannot* assess the accuracy regarding the target size and angle for the 9DoF 3D bounding boxes.

3D SOT Tasks. GSOT3D consists of sequences of multiple modalities, comprising *point cloud*, *RGB image*, and *depth*. This allows research on various 3D tracking tasks, including the single-modal *3D SOT on point cloud (PC)* **3D-SOT_{PC}**, and multi-modal *3D SOT on RGB-PC* (**3D-SOT_{RGB-PC}**) and *3D SOT on RGB-D* (**3D-SOT_{RGB-D}**).

Given the initial 3D target box, 3D-SOT_{PC} aims to locate the target on the point clouds (see Fig. 4 (a)), 3D-SOT_{RGB-PC} localizes target object with point clouds and RGB images (see Fig. 4 (b)), aiming to enhance the 3D tracking through appearance, and 3D-SOT_{RGB-D} focuses on localizing the target using RGB and depth images (see Fig. 4 (c)), providing a more cost-effective solution for 3D tracking. Due to limited space, please refer to our **supplementary material** for the detailed formulation of these tasks.

For all tasks, except for used modalities, the dataset split

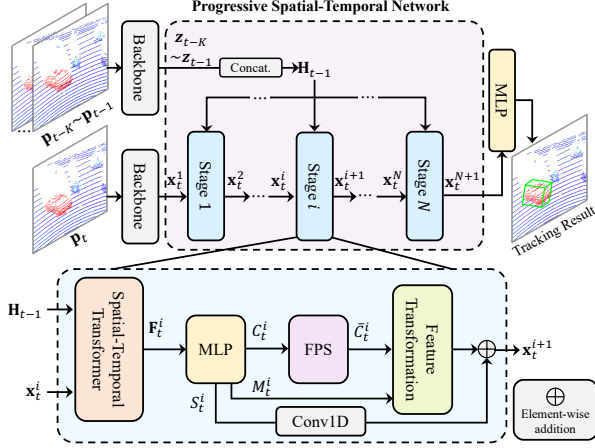


Figure 5. Architecture of the proposed PROT3D.

and evaluation metric are the same. Please **note**, since there are **very few** trackers for 3D-SOT_{RGB-PC} and 3D-SOT_{RGB-D}, we primarily focus on 3D-SOT_{PC} in later baseline design and experiments due to more available trackers, and leave the study on SOT_{RGB-PC} and 3D-SOT_{RGB-D} to future work.

4. The Proposed PROT3D

We present a simple yet effective tracker, PROT3D, for 3D-SOT_{PC}, as there are more available trackers for SOT_{PC}, and we will explore 3D-SOT_{RGB-PC} and 3D-SOT_{RGB-D} in the future. The key is to *progressively* refine search region feature with multiple cascaded stages, as in Fig. 5. Each stage performs spatial-temporal target localization, and the result is used to augment the search region feature in the next stage.

Similar to [28], PROT3D treats 3D tracking as a matching problem. Inspired by [37], we leverage target cues from historical frames for robust performance. More specifically, given point cloud \mathbf{p}_t at frame t , we apply information from previous K frames $\{\mathbf{p}_j\}_{j=t-K}^{t-1}$ for tracking. We first extract their features through a shared backbone $\Phi(\cdot)$ as follows,

$$\mathbf{x}_t^1 = \Phi(\mathbf{p}_t) \quad \mathbf{z}_j = \Phi(\mathbf{p}_j) \quad j = t-K, \dots, t-1 \quad (1)$$

where \mathbf{x}_t^1 represents the feature of \mathbf{p}_t and \mathbf{z}_j is the feature of \mathbf{p}_j ($j = t-K, \dots, t-1$). Then, we concatenate all features from historical frames via $\mathbf{H}_{t-1} = \text{concat}(\mathbf{z}_{t-K}, \dots, \mathbf{z}_{t-1})$ to obtain memory feature \mathbf{H}_{t-1} for frame t . After that, \mathbf{H}_{t-1} and \mathbf{x}_t^1 are sent to the progressive spatial-temporal network with multiple stages, with each performing localization.

Specifically, for stage i , it receives \mathbf{H}_{t-1} and \mathbf{x}_t^i as inputs. Then, a spatial-temporal Transformer is utilized to fuse the memory \mathbf{H}_{t-1} into \mathbf{x}_t^i , as follows

$$\mathbf{F}_t^i = \text{SPT}(\mathbf{x}_t^i, \mathbf{H}_{t-1}) \quad (2)$$

where \mathbf{F}_t^i is the feature after fusion. $\text{SPT}(\cdot, \cdot)$ represents the spatial-temporal Transformer, and comprises L (L is set to 2) layers. Similar to [37], each layer consists of cross- and self-attention operations [30] and a feed-forward network,

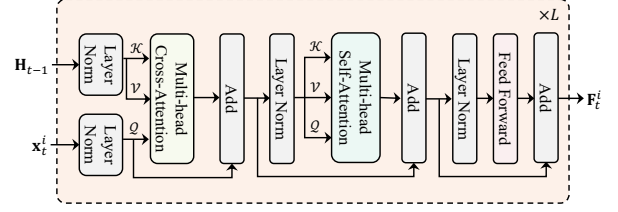


Figure 6. Architecture of spatial-temporal Transformer.

as displayed in Fig. 6. After that, \mathbf{F}_t^i is forwarded to a multi-layer perceptron (MLP) for localization, as follows

$$\mathbf{R}_t^i = \text{MLP}(\mathbf{F}_t^i) \quad (3)$$

where $\mathbf{R}_t^i = [C_t^i, M_t^i, S_t^i]$ is the localization result, with C_t^i potential target center, M_t^i targetness mask, and S_t^i proposal scores. Then, we perform Farthest Point Sampling (FPS) on C_t^i to refine point clouds, as follows

$$\bar{C}_t^i = \text{FPS}(C_t^i) \quad (4)$$

where \bar{C}_t^i is sampled points. After FPS, the \bar{C}_t^i and M_t^i are fed to a feature transformation block (FTB) and the resulted feature is combined with the score information to generate the refined search region feature \mathbf{x}_t^{i+1} , mathematically described as follows,

$$\mathbf{x}_t^{i+1} = \text{FTB}(\bar{C}_t^i, M_t^i) + \text{Conv1D}(S_t^i) \quad (5)$$

where $\text{FTB}(\cdot, \cdot)$ is feature transformation block, borrowed from [37], and contains point-to-reference and a 3D convolution operation (see **supplementary material** for details). $\text{Conv1D}(\cdot)$ is 1D convolution to embed S_t^i to score feature.

Please note, \mathbf{x}_t^{i+1} in Eq. (5) is generated by encoding target information C_t^i , M_t^i , and S_t^i , obtained via localization, and thus more discriminative for distinguishing target from background. For further refinement, \mathbf{x}_t^{i+1} is fed to the next stage ($i+1$), forming a progressive cascade architecture. This way, the search region feature can be gradually refined with more target cues, benefiting the final localization.

After the last N^{th} stage, the generated \mathbf{x}_t^{N+1} is employed for final 9DoF target localization via MLP, as follows,

$$\mathcal{R}_t = \text{MLP}(\mathbf{x}_t^{N+1}) \quad (6)$$

where $\mathcal{R}_t = [\mathcal{B}_t, \mathcal{S}_t] \in \mathbb{R}^{D \times 10}$, with $\mathcal{B}_t \in \mathbb{R}^{D \times 9}$ the 9DoF box parameters, $\mathcal{S}_t \in \mathbb{R}^{D \times 1}$ the targetness scores and D the number of points in \mathbf{x}_t^{N+1} . Finally, the tracking result b_t is determined as follows,

$$b_t = \mathcal{B}_t(h) \quad \text{where } h = \arg \max_{d=1, \dots, D} \mathcal{S}(d) \quad (7)$$

where $b_t = (x_t^*, y_t^*, x_t^*, \alpha_t^*, \beta_t^*, \gamma_t^*, l_t^*, h_t^*, w_t^*)$, predicting the translation offset (x_t^*, y_t^*, x_t^*) of the center point and angle offset $(\alpha_t^*, \beta_t^*, \gamma_t^*)$ and size offset (l_t^*, h_t^*, w_t^*) of target box from frame $(t-1)$ to frame t .

Please note, PROT3D is a *class-agnostic* 3D tracker that is able to track the target object of any categories. The loss of PROT3D is computed with loss function for final target

Table 3. Overall performance of eight state-of-the-art trackers and our PROT3D on 3D-SOT_{PC} using mAO, mSR₅₀, and mSR₇₅. The best three results are highlighted in red, blue, and green fonts, respectively. Our PROT3D achieves the best results on all three metrics.

		P2B [28]	BAT [39]	PTT [29]	M2-Track [40]	CXTrack [36]	MBPTrack [37]	SeqTrack- 3D [21]	M3SOT [22]	PROT3D (ours)
w/ training on GSOT3D	mAO (%)	9.79	6.56	14.00	20.26	14.29	20.54	8.61	17.40	21.97
	mSR ₅₀ (%)	8.59	3.54	10.42	14.34	8.39	16.55	5.25	12.47	19.76
	mSR ₇₅ (%)	1.75	0.88	1.60	1.88	1.02	2.57	1.11	1.74	5.22
w/o training on GSOT3D	mAO (%)	2.81	1.91	2.36	3.65	2.42	3.38	1.54	2.68	-
	mSR ₅₀ (%)	1.35	1.24	1.29	1.32	1.19	1.81	0.90	1.36	-
	mSR ₇₅ (%)	0.60	0.60	0.67	0.61	0.63	0.65	0.61	0.62	-

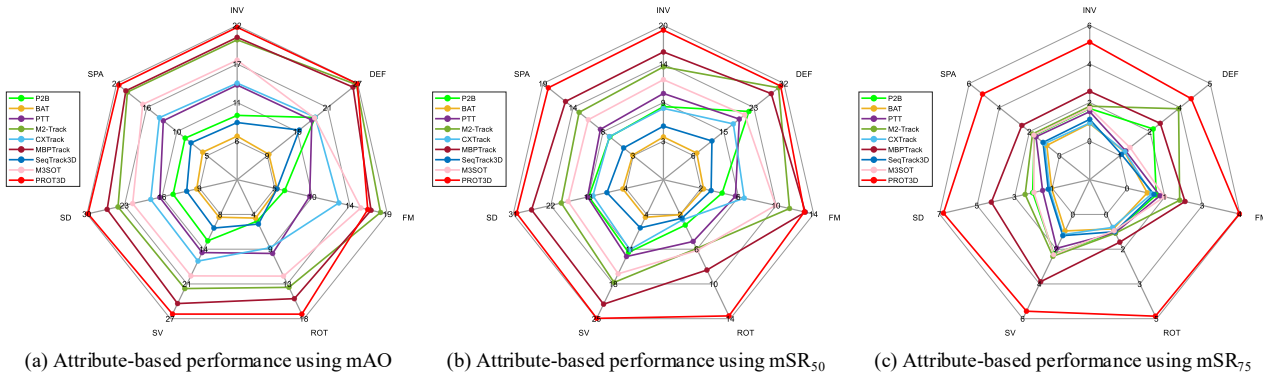


Figure 7. Attribute-based performance and comparison using mAO (image (a)), mSR₅₀ (image (b)), and mSR₇₅.

estimation. Due to space limitation, please refer to our **supplementary material** for details of the loss function.

Implementation. PROT3D is implemented using PyTorch [26], and trained for 80 epochs using Adam [17]. The initial learning rate is 0.001, and the batchsize is 9. In PROT3D, the number of stages is set to 2, and the memory size K is set to 3. Our full code and model will be released.

5. Experiments

Please **note** again, we primary focus on experiments for 3D-SOT_{PC} trackers, as most currently open-sourced 3D trackers with available implementations belong to 3D-SOT_{PC}.

Evaluated Trackers. We evaluate eight representative 3D trackers that share their executable codes on GSOT3D, and provide basis for the future comparison, including P2B [28], BAT [39], PTT [29], M2-Track [40], CXTrack [36], MBPTrack [37], SeqTrack3D [21], and M3SOT [22]. The summary of these trackers is in the **supplementary material**.

5.1. Evaluation Results

Overall Performance. We evaluate eight representative 3D trackers on 3D-SOT_{PC} and the proposed PROT3D on test set of GSOT3D. Tab. 3 displays the results and comparison using mAO, mSR₅₀, and mSR₇₅. For the fair comparison, we retrain all evaluated trackers using training set of GSOT3D and compare them with our PROT3D in the Tab. 3. We can observe that, PROT3D achieves the best result with 21.97% mAO, 19.76% mSR₅₀, and 5.22% mSR₇₅, outperforming

the second best MBPTrack with 20.54% mAO by 1.43%, 16.55% mSR₅₀ by 3.21%, and 2.57% mSR₇₅ by 2.65% and the third best M2-Track with 20.26% mAO by 1.71%, 14.34% mSR₅₀ by 5.42, and 1.88% mSR₇₅ by 3.34%. This evidences the superiority of PROT3D with progressive refinement for more robust generic tracking. It is worth noting that, for all trackers, the mSR₇₅ score is much lower than the mSR₅₀ score, as mSR₇₅ has a higher threshold (0.75) than mSR₅₀ (0.5) and thus is more restrict.

Besides, Tab. 3 shows comparison of evaluated trackers using GSOT3D_{Tra} or not for retraining. For the tracker that does not use GSOT3D_{Tra} for training, we directly utilize its default model pre-trained from KITTI for evaluation. As in Tab. 3, we observe that, re-training these trackers on GSOT3D can significantly improve their results on all three metrics. This shows the necessity of a more diverse dataset such as our GSOT3D for generic 3D object tracking.

Attribute-based Performance. In order to further analyze different algorithms, we conduct evaluation and comparison under seven attributes using mAO, mSR₅₀, and mSR₇₅. For fair comparison, all the compared trackers are trained using GSOT3D_{Tra}. Fig. 7 reports the results. From Fig 7, we can see that, the proposed PROT3D achieves the best results on six out of seven attributes using mAO and mSR₅₀, and the best results on all seven attributes on all seven attributes using harder mSR₇₅. All these results show that, PROT3D is more robust and precise than other trackers in tracking.

Because of limited space, we demonstrate more qualitative results and analysis in the **supplementary material**.

Table 4. Comparison of GSOT3D with KITTI.

	KITTI [11]			GSOT3D (ours)		
	mAO (%)	mSR ₅₀ (%)	mSR ₇₅ (%)	mAO (%)	mSR ₅₀ (%)	mSR ₇₅ (%)
P2B [28]	63.25	78.57	39.52	9.79	8.59	1.75
BAT [39]	56.65	70.44	32.70	6.56	3.54	0.88
PTT [29]	52.30	66.32	40.79	14.00	10.42	1.60
M2-Track [40]	67.71	86.43	44.00	20.26	14.34	1.88
CXTrack [36]	70.18	87.95	46.06	14.29	8.39	1.02
MBPTrack [37]	71.95	90.50	51.54	20.54	16.55	2.57
SeqTrack3D [21]	32.01	32.28	11.36	8.61	5.25	1.11
M3SOT [22]	64.58	81.33	35.38	17.40	12.47	1.74

5.2. Comparison with Other Benchmark

KITTI [11] is currently the most popular dataset for 3D SOT on point clouds. Nevertheless, as mentioned before, the sequences of KITTI are limited to only a few object categories and constrained traffic scenarios, making it not suitable for generic 3D object tracking. Compared to KITTI, GSOT3D includes more target classes from diverse environments. As a consequence, our GSOT3D is more challenging but realistic for real-world applications.

We conduct a comparison of our GSOT3D with KITTI. Tab. 4 reports the results of evaluated trackers on GSOT3D and KITTI using mAO, mSR₅₀, and mSR₇₅. As shown in Tab. 4, we clearly see that, all current trackers suffer from a significant performance drop on GSOT3D, which shows the challenges from more categories and diverse scenarios and more efforts are needed for generic 3D object tracking.

5.3. Ablation Study on PROT3D

9DoF box prediction and progressive architecture. Different from previous 3D trackers predicting a 7DoF bounding box, our PROT3D estimates a more precise 9DoF 3D bounding box as the tracking result. In addition, PROT3D applies a novel progressive architecture for tracking, which enables better features for robust localization. Tab. 5 lists the experiment results. The baseline (❶) contains one stage and predicts a 7DoF box, and achieves the mAO of 19.86%, mSR₅₀ of 15.16%, and mSR₇₅ of 2.36%. When changing to the 9DoF box prediction (❷), the performance is improved to 20.03% mAO, 15.46% mSR₅₀, and 3.29% mSR₇₅, showing effectiveness of using 9DoF for 3D tracking. It is worth noting, the gains by 9DoF are not very significant, as most objects in GSOT3D are rigid and only a small part of the sequences contain deformable objects. Nonetheless, in the real world, there exist more non-rigid objects, and 9DoF box prediction is still more desirable. When further applying our progressive architecture (❸), the results are largely boosted to 21.97% mAO, 19.76% mSR₅₀, 5.22% mSR₇₅, which clearly validates the efficacy of our progressive refinement for generic 3D object tracking.

Number of progressive stages. The core of our PROT3D

Table 5. Analysis of 9DoF prediction and progressive architecture

	9DoF Box	Progressive Architecture	mAO (%)	mSR ₅₀ (%)	mSR ₇₅ (%)
❶	-	-	19.86	15.16	2.36
❷	✓	-	20.03	15.46	3.29
❸	✓	✓	21.97	19.76	5.22

Table 6. Analysis of the number N of stages in our PROT3D.

	Number of Stages	mAO (%)	mSR ₅₀ (%)	mSR ₇₅ (%)
❶	$N = 1$	20.03	15.46	3.29
❷	$N = 2$	21.97	19.76	5.22
❸	$N = 3$	21.58	19.61	5.19

Table 7. Analysis of the memory size K in our PROT3D.

	Memory Size	mAO (%)	mSR ₅₀ (%)	mSR ₇₅ (%)
❶	$K = 2$	21.37	19.52	5.32
❷	$K = 3$	21.97	19.76	5.22
❸	$K = 4$	21.84	19.69	5.17

is a progressive network with multiple stages of refinement. To explore the impact of number N of stages in PROT3D, we conduct an ablation in Tab. 6. We observe, when using two stages (❷), PROT3D shows the best results of 21.97% mAO, 19.76 mSR₅₀, and 5.22% mSR₇₅. When further increasing the number of stages to 3 (❸), the performance is slightly decreased. Thus, we set N to 2 in this work.

Memory size. We adopt a memory containing previous K frames for tracking. We ablate the memory size K in Tab. 7. We observe that, when using 3 previous frames (❷) in the memory, PROT3D shows the best tracking performance.

6. Conclusion and Limitation

In this paper, we introduce GSOT3D, a new benchmark for generic 3D SOT. It contains 620 multimodal sequences with over 123K frames, and supports different 3D single object tracking tasks. To the best of our knowledge, GSOT3D is the largest benchmark to date dedicated to 3D SOT. Besides, we assess several representative trackers on GSOT3D to understand their performance and to offer comparison for future research. Furthermore, we present a simple yet effective progressive tracker PROT3D and obtain state-of-the-art result. We believe that, our benchmark, evaluation, and new baseline will inspire more research towards generic 3D object tracking and facilitate its real-world applications.

Despite contributions, there exist a few limitations. First, the experiments are mainly focused on the 3D-SOT_{PC}, and study on 3D-SOT_{RGB-PC} and 3D-SOT_{RGB-D} is not provided. Second, the sequences in GSOT3D are relatively short, and not suitable for long-term tracking. Given 3D-SOT_{PC} is the current research focus and our major goal is to offer a new benchmark for generic tracking, we leave study of more 3D tracking tasks and long-term 3D tracking to the future work.

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 More Statistics

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

S4 Evaluation Metrics and 3D IoU

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

S5 Formulation of Different 3D SOT Tasks

We describe the formulation of different 3D SOT tasks.

S6 Details of Feature Transformation Block

We present the details of the feature transformation block adopted in our PROT3D.

S7 Loss Function

We present details of the loss function to train PROT3D.

S8 Summary of Evaluated Trackers

We offer a summary for trackers assessed on GOST3D.

S9 Qualitative Results

We offer more qualitative analysis of our PROT3D and its comparison to other trackers on GSOT3D.

S10 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 [6]. Fig. 8 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. 8.

S2 Annotation Tool

For data labeling, we use the annotation tool provided by a company. Fig. 9 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

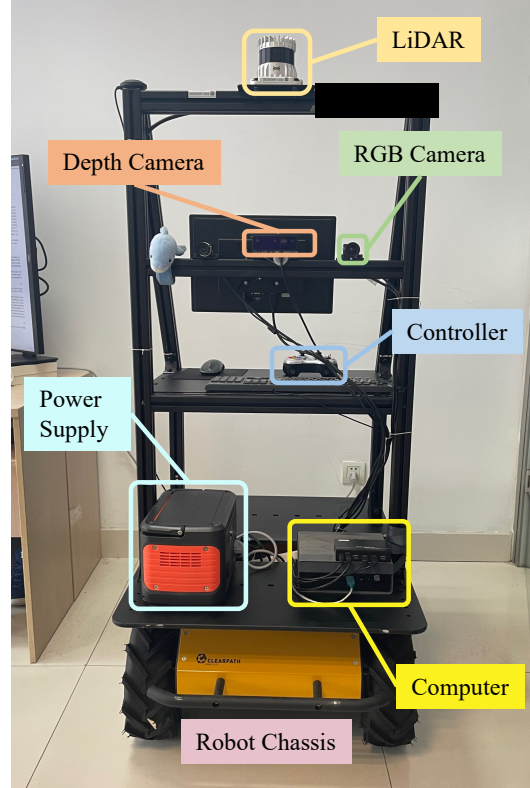


Figure 8. Our mobile robotic platform for data collection.

Table 8. 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

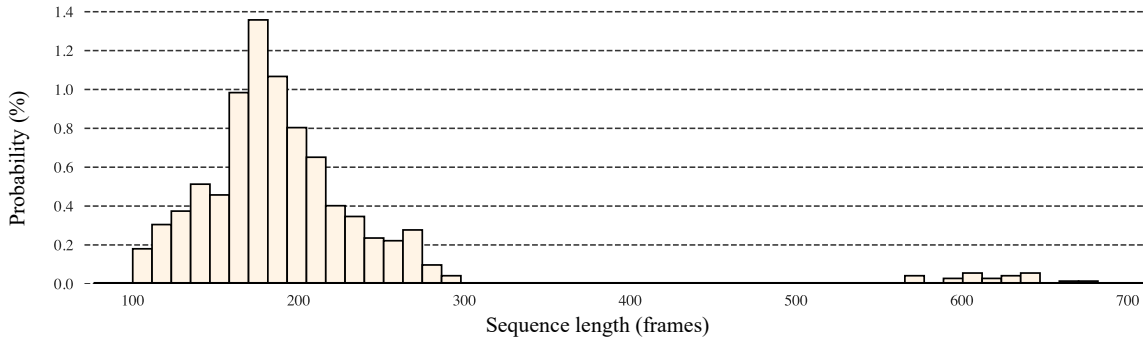
be flexibly zoomed in or out). Then, the initial 3D bounding box is refined by adjusting the 2D boxes on each projected view on XY, XZ, and YZ planes. In the annotation tool, a preview of the 3D box in the RGB image is provided for visual inspection of the refined box. By doing this, we can ensure the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

S3 More Statistics

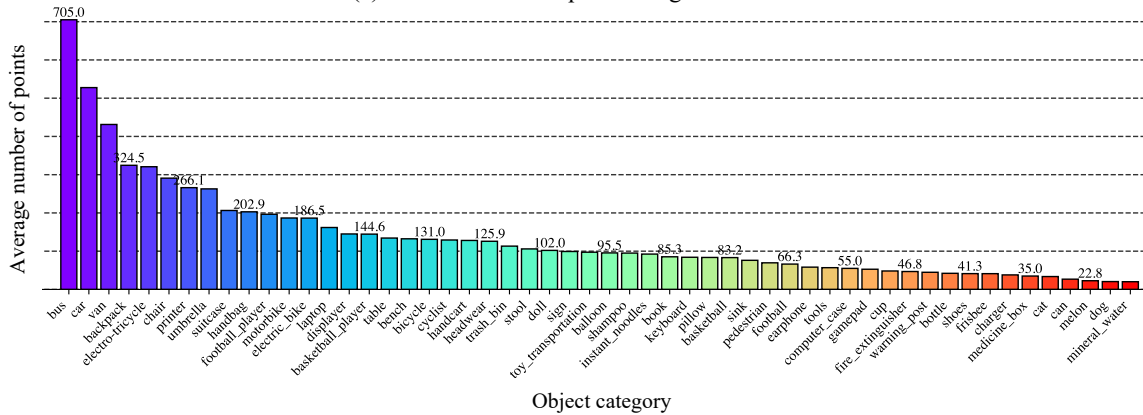
In this section, we demonstrate more statistics of GSOT3D. In specific, Fig. 10 (a) shows distribution of sequence length on GSOT3D. Although the average length of GSOT3D is 198 frames, there exist several relatively longer ones with sequence length larger than 600 frames, which can be used for analyzing trackers on relatively longer sequences. Be-



Figure 9. Annotation interface of our used annotation tool.



(a) Distribution of sequence length on GSOT3D



(b) Average number of points in each object category

Figure 10. Statistics on GSOT3D. Image (a): Distribution of sequence length. Image (b): Average number of points in each object category

sides, Fig. 10 (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.

S4 Evaluation Metrics and 3D IoU

Inspired by [14], we utilize mean Average Overlap (**mAO**) and mean Success Rate (**mSR**) to measure different tracking algorithms. Specifically, mAO is calculated by averaging the class-wise overlaps, *i.e.*, 3D Intersection over Union (3D IoU, which will be detailed 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,

$$\begin{aligned} \text{mAO} &= \frac{1}{C} \sum_{c=1}^C \left(\frac{1}{|S_c|} \sum_{i \in S_c} \text{AO}_i \right) \\ \text{mSR} &= \frac{1}{C} \sum_{c=1}^C \left(\frac{1}{|S_c|} \sum_{i \in S_c} \text{SR}_i \right) \end{aligned} \quad (8)$$

where C is the total number of object categories in GSOT3D, S_c the set of all sequences belonging to category c . AO_i represents the Average Overlap (AO) for the i^{th} sequence in S_c , and SR_i denotes Success Rate (SR). mSR_{50} and 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 the targets that have symmetric structure. Nevertheless, in our GSOT3D, there exist many targets with symmetric structure, such as *ball*, *umbrella*, and so forth (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 leverage the strategy employed in [1, 4] 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 [7].

Therefore, for non-symmetric targets, we use method as in KITTI [11] for 3D IoU calculation, while for symmetric targets, we use strategy as in [1, 4] for 3D IoU computation.

S5 Formulation of Different 3D SOT Tasks

GSOT3D is a unique platform to broaden research direction in 3D SOT by supporting different tasks, comprising single-modal 3D object tracking, *i.e.*, *3D SOT on Point Cloud (PC)* (3D-SOT_{PC}), and multi-modal 3D tracking, *i.e.*, *3D SOT on RGB-PC* (3D-SOT_{RGB-PC}) or *RGB-Depth* (3D-SOT_{RGB-D}).

3D-SOT_{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 pro-

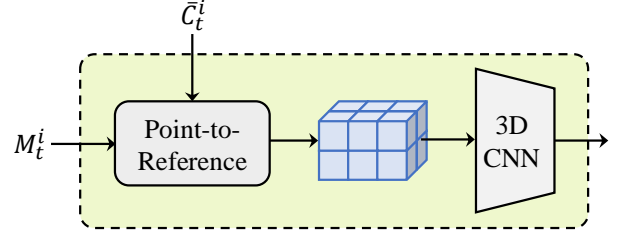


Figure 11. Architecture of the feature transformation block.

cess can be formulated as follows,

$$\{b_i\}_{i=2}^N \leftarrow \mathcal{T}_{\text{PC}}(\{\mathbf{p}_i\}_{i=1}^N, b_1) \quad (9)$$

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}_{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_{PC}, **3D-SOT_{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,

$$\{b_i\}_{i=2}^N \leftarrow \mathcal{T}_{\text{RGB-PC}}(\{\mathbf{p}_i\}_{i=1}^N, \{I_i\}_{i=1}^N, b_1) \quad (10)$$

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_{RGB-D}** exploits a more economic way using RGB and depth images for 3D tracking, and can be formulated as follows,

$$\{b_i\}_{i=2}^N \leftarrow \mathcal{T}_{\text{RGB-D}}(\{D_i\}_{i=1}^N, \{I_i\}_{i=1}^N, b_1) \quad (11)$$

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. (10).

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

S6 Details of Feature Transformation Block

Fig. 11 displays feature transformation block (FTB) applied in each stage of our PROT3D. The feature transformation block is borrowed from [37] 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 [32] for feature aggregation, as follows,

$$\begin{aligned} \hat{g}_t^i &= \text{Point-to-Reference}(\bar{C}_t^i, M_t^i) \\ &= \text{EdgeConv}(\text{MLP}(\text{Concatenate}(\bar{C}_t^i, M_t^i))) \end{aligned} \quad (12)$$

After this, the resulted feature \hat{g}_t^i is fed into a 3D CNN network to generate point-wise feature. Fig. 11 illustrates FTB. For more details, please kindly refer to [37].

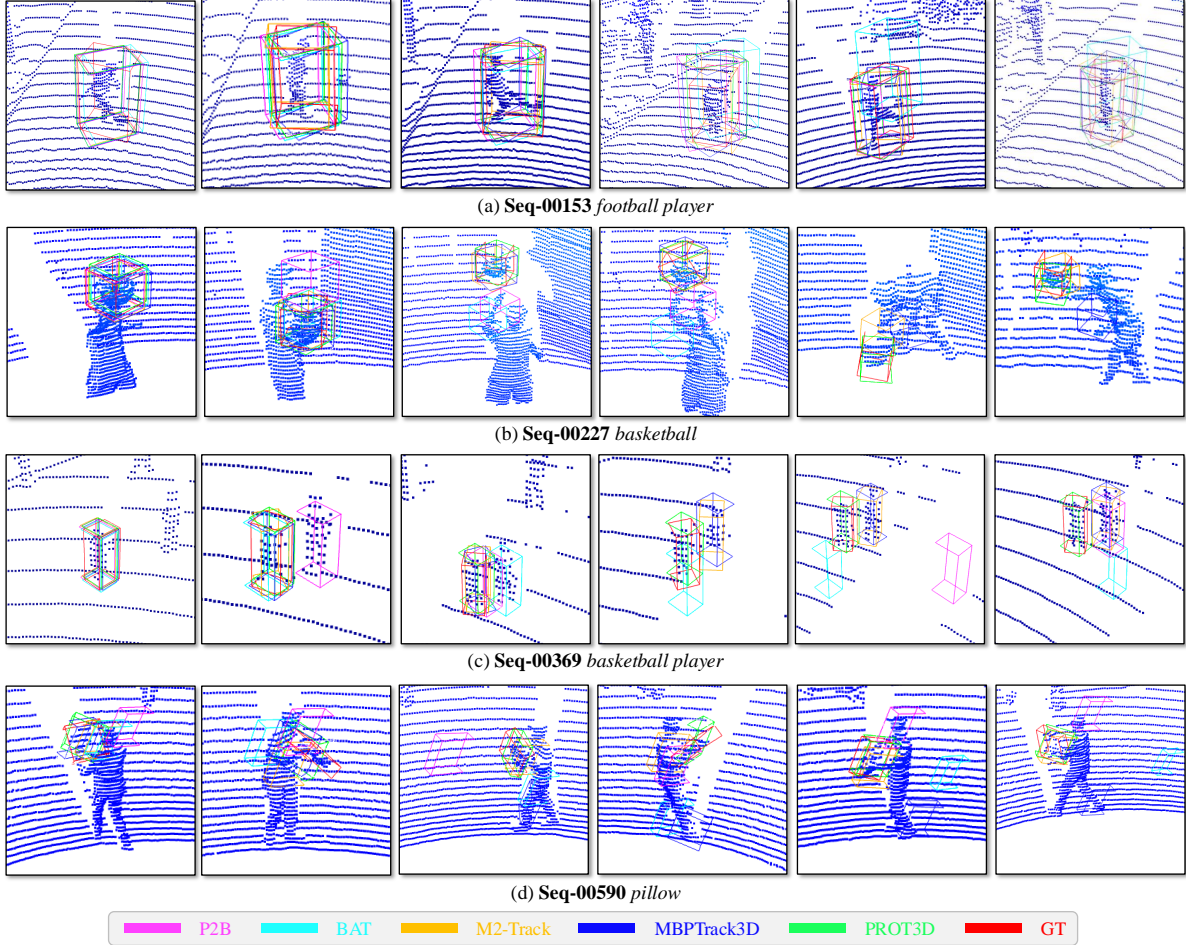


Figure 12. Qualitative results of several evaluated trackers and our proposed PROT3D. We can see that, the proposed PROT3D locates target object in different scenarios, showing its robustness for generic 3D object tracking.

S7 Loss Function

In this section, we present details regarding the loss function for training PROT3D. Specifically, after the N^{th} stage, the final feature \mathbf{x}_t^{N+1} is sent to the MLP layer for prediction. Similar to previous work [37], we use the following loss function for end-to-end training,

$$\mathcal{L}_{\text{total}} = \lambda_m \mathcal{L}_m + \lambda_c \mathcal{L}_c + \lambda_p \mathcal{L}_p + \lambda_s \mathcal{L}_s + \mathcal{L}_{\text{bbox}} \quad (13)$$

where $\mathcal{L}_{\text{total}}$ represents the total training loss, \mathcal{L}_m the standard cross-entropy loss to supervise the targetness mask, \mathcal{L}_c the mean square loss to supervise the target center, \mathcal{L}_p the cross-entropy loss to supervise proposal score, \mathcal{L}_s the cross-entropy loss to supervise the targetness score \mathcal{S}_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). λ_m , λ_c , λ_p , λ_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.

Table 9. Summary of evaluated trackers on GSOT3D.

Tracker	Where	Backbone	Transformer
P2B [28]	CVPR'20	PointNet++	✗
BAT [39]	ICCV'21	PointNet++	✗
PTT [29]	IROS'21	PointNet++	✓
M2-Track [40]	CVPR'22	PointNet	✗
CXTrack [36]	CVPR'23	DGCNN	✓
MBPTrack [37]	ICCV'23	DGCNN	✓
SeqTrack3D [21]	ICRA'24	PointNet++	✓
MS3SOT [22]	AAAI'24	DGCNN	✓

S8 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 [28], BAT [39], PTT [29], M2-Track [40], CXTrack [36], MBPTrack [37], SeqTrack3D [21], and M3SOT [22]. 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. 9 summarizes these trackers.

S9 Qualitative Results

In this section, we show qualitative results of different trackers and our PROT3D on GSOT3D in Fig. 12. From Fig. 12, 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.

S10 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. 11
- [2] Alireza Asvadi, Pedro Girao, Paulo Peixoto, and Urbano Nunes. 3d object tracking using rgb and lidar data. In *ITSC*, 2016. 1
- [3] Adel Bibi, Tianzhu Zhang, and Bernard Ghanem. 3d part-based sparse tracker with automatic synchronization and registration. In *CVPR*, 2016. 1, 3
- [4] 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. 11
- [5] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multi-modal dataset for autonomous driving. In *CVPR*, 2020. 1, 2, 3
- [6] Ankit Dhall, Kunal Chelani, Vishnu Radhakrishnan, and K Madhava Krishna. Lidar-camera calibration using 3d-3d point correspondences. *arXiv*, 2017. 9
- [7] Christer Ericson. *Real-time collision detection*. Crc Press, 2004. 11
- [8] 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. 3, 4
- [9] 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. 4
- [10] Hamed Kiani Galoogahi, Ashton Fagg, Chen Huang, Deva Ramanan, and Simon Lucey. Need for speed: A benchmark for higher frame rate object tracking. In *ICCV*, 2017. 3
- [11] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 1, 2, 3, 8, 11
- [12] Silvio Giancola, Jesus Zarzar, and Bernard Ghanem. Leveraging shape completion for 3d siamese tracking. In *CVPR*, 2019. 1, 3
- [13] Zhiyang Guo, Yunyao Mao, Wengang Zhou, Min Wang, and Houqiang Li. Cmt: Context-matching-guided transformer for 3d tracking in point clouds. In *ECCV*, 2022. 3
- [14] 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, 5, 11
- [15] Le Hui, Lingpeng Wang, Mingmei Cheng, Jin Xie, and Jian Yang. 3d siamese voxel-to-bev tracker for sparse point clouds. In *NeurIPS*, 2021. 3
- [16] Le Hui, Lingpeng Wang, Linghua Tang, Kaihao Lan, Jin Xie, and Jian Yang. 3d siamese transformer network for single object tracking on point clouds. In *ECCV*, 2022. 3
- [17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 7
- [18] Matej Kristan, Jiri Matas, Aleš Leonardis, Tomáš Vojtíš, Roman Pflugfelder, Gustavo Fernandez, Georg Nebehay, Fatih Porikli, and Luka Čehovin. A novel performance evaluation methodology for single-target trackers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11):2137–2155, 2016. 3
- [19] Annan Li, Min Lin, Yi Wu, Ming-Hsuan Yang, and Shuicheng Yan. Nus-pro: A new visual tracking challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):335–349, 2015.
- [20] Pengpeng Liang, Erik Blasch, and Haibin Ling. Encoding color information for visual tracking: Algorithms and benchmark. *IEEE Transactions on Image Processing*, 24(12):5630–5644, 2015. 3
- [21] Yu Lin, Zhiheng Li, Yubo Cui, and Zheng Fang. Seqtrack3d: Exploring sequence information for robust 3d point cloud tracking. In *ICRA*, 2024. 7, 8, 12
- [22] 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. 7, 8, 12

- [23] Teli Ma, Mengmeng Wang, Jimin Xiao, Huifeng Wu, and Yong Liu. Synchronize feature extracting and matching: A single branch framework for 3d object tracking. In *ICCV*, 2023. 3
- [24] Matthias Muller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *ECCV*, 2018. 3
- [25] Jiahao Nie, Zhiwei He, Xudong Lv, Xueyi Zhou, Dong-Kyu Chae, and Fei Xie. Towards category unification of 3d single object tracking on point clouds. In *ICLR*, 2024. 1, 3
- [26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 7
- [27] Liang Peng, Junyuan Gao, Xinran Liu, Weihong Li, Shaohua Dong, Zhipeng Zhang, Heng Fan, and Libo Zhang. Vast-track: Vast category visual object tracking. In *NeurIPS*, 2024. 3
- [28] 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. 1, 3, 6, 7, 8, 12
- [29] 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. 3, 7, 8, 12
- [30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017. 3, 6
- [31] Xiao Wang, Xiujun Shu, Zhipeng Zhang, Bo Jiang, Yaowei Wang, Yonghong Tian, and Feng Wu. Towards more flexible and accurate object tracking with natural language: Algorithms and benchmark. In *CVPR*, 2021. 3
- [32] 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. 11
- [33] Qiangqiang Wu, Yan Xia, Jia Wan, and Antoni B Chan. Boosting 3d single object tracking with 2d matching distillation and 3d pre-training. In *ECCV*, 2023. 3
- [34] Qiao Wu, Kun Sun, Pei An, Mathieu Salzmann, Yanning Zhang, and Jiaqi Yang. 3d single-object tracking in point clouds with high temporal variation. In *ECCV*, 2024. 1, 3
- [35] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *CVPR*, 2013. 3
- [36] 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. 1, 3, 7, 8, 12
- [37] 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. 1, 3, 6, 7, 8, 11, 12
- [38] 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. 2, 3, 4
- [39] 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. 1, 3, 7, 8, 12
- [40] 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. 3, 7, 8, 12
- [41] Changqing Zhou, Zhipeng Luo, Yueru Luo, Tianrui Liu, Liang Pan, Zhongang Cai, Haiyu Zhao, and Shijian Lu. Pptr: Relational 3d point cloud object tracking with transformer. In *CVPR*, 2022. 3