2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Self-Supervised Point Cloud Single Object Tracking

Yuheng Liu[1], Le Hui[2,3,†], Ziyue Zhu[1], Shaohui Mei[2], Yigong Zhang[1], Jin Xie[4,†] and Jian Yang[1,4,†]

***Abstract*— Point cloud single object tracking is critical in autonomous driving. However, current methods heavily rely on frame-by-frame human annotations, which do not scale well with the growing amount of unlabeled LiDAR data. In this paper, we propose the first self-supervised point cloud single object tracking framework, eliminating the need for any manual labels. Our method integrates motion, geometry, and semantic cues to generate plausible object proposals and tracks the target using a predictive filter. Specifically, we generate pseudo labels by clustering local motion patterns from scene flow, while pre-training a proposal network using point cloud forecasting as a proxy task to learn global motion patterns and geometric shape priors. Then, we train the proposal network using the initial pseudo labels and iteratively refine them by treating semantic features as evolving prototypes in each training round. Finally, a simple motion filter is employed to predict the target's current state based on its past dynamics. Evaluated on KITTI, nuScenes, and Waymo, our self-supervised point cloud single object tracking approach is on par with—and in some cases outperforms—fully supervised trackers, demonstrating that self-supervision is a scalable path forward for 3D single object tracking.**

I. Introduction

Single object tracking (SOT) in point clouds is fundamental for autonomous driving perception [1], [2], [3], [4]. While the proliferation of LiDAR sensors has made large-scale data accessible [5], [6], [7], the reliance on expensive frame-by-frame manual annotations limits the scalability of supervised methods. This necessitates the development of effective label-free tracking approaches.

SOT is commonly addressed by appearance matching through Siamese networks developed from 2D image tracking [8], [9]. Building on this, self-supervised SOT methods for camera inputs often combine Siamese architectures with a "cycle training" strategy [10], [11], [12], as shown in Fig. 1(a). These approaches typically generate a pseudo ground truth in the first frame using optical flow, then train the network forward and backward (e.g, "0->1->2->1->0") to obtain consistent supervision signals.

While most supervised SOT methods have extended from 2D image to 3D point cloud [13], [14], [15], [16], [17], [18], [19], [20], [21], adapting the "cycle training" strategy for self-supervised point cloud SOT faces two key challenges:

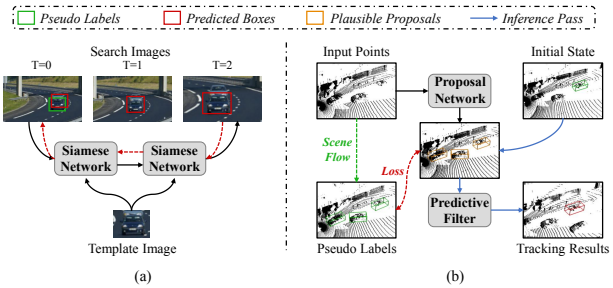

Fig. 1: Comparison between current self-supervised 2D SOT methods and our self-supervised point cloud single object tracking (SPSOT) framework. (a) Siamese-based "cycle training" for 2D images. (b) Our SPSOT framework.

1) Unlike dense image features, point clouds are sparse and textureless, making appearance-based matching unreliable without ground-truth labels. 2) End-to-end Siamese training lacks intermediate supervision and fails when initial pseudo labels are poor, which is a frequent issue in sparse 3D data (e.g., ⅓ of Waymo objects have less than 38 points).

To address these issues, we propose a self-supervised point cloud single object tracking (SPSOT) framework, illustrated in Fig. 1(b). Instead of relying on a fixed target location in the first frame, our method generates multiple proposals per frame and employs a predictive motion filter to trace the target trajectory. This approach is motivated by two key observations: 1) While appearance features are unreliable without labels, 3D motion is governed by physical laws and can be robustly predicted. A simple motion filter can easily discard implausible proposals. 2) This setup provides frame-by-frame supervisory signals, making it easier to train a proposal network than to train a full appearance-based tracker.

Specifically, we first cluster scene flow vectors to generate pseudo labels, akin to how optical flow is used in 2D self-supervised SOT [10], [11]. These pseudo labels are used to train a proposal network that aims to "discover" plausible target locations. However, we find that critical motion cues exist only locally during clustering (i.e., nearby points with similar motion are grouped as an object), and these local dynamics are lost when the bounding boxes are passed to the proposal network. We argue that broader, global motion patterns should not be overlooked, as they reveal contextual relationships between objects. To this end, we pre-train our feature extractor using point cloud forecasting [22], [23], a proxy task that captures temporal motion across the scene. Inspired by the success of volume rendering in point forecasting [24], [25], we incorporate ray-casting into this pre-

[1] PCA Lab, VCIP, College of Computer Science, Nankai University, Tianjin, China; {liuyuhengnk, zhuziyue}@mail.nankai.edu.cn, {zyg025, csjyang}@nankai.edu.cn
[2] Electronics and Information, Northwestern Polytechnical University, Xi'an, China; {huile, meish}@nwpu.edu.cn
[3] Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China
[4] School of Intelligence Science and Technology, Nanjing University, Suzhou, China; csjxie@nju.edu.cn
† Corresponding authors.

training to enhance geometric understanding. To further refine the proposal network, we feed high-level semantic features back into the point cloud, treating them as discriminative prototypes for re-clustering points with similar motion. This enables an iterative refinement of the pseudo labels.

Our contributions are summarized as follows:

- We propose the first fully self-supervised point cloud single object tracking framework, unifying motion, geometry, and semantic information without human annotations.
- We treat scene flows as local motion embeddings for pseudo label generation and use point cloud forecasting as a proxy task for learning global motion and geometry structure awareness.
- We introduce semantic prototypes into the clustering process to iteratively refine pseudo labels and improve proposal network training.

II. Related works

A. 2D Self-supervised single object tracking

Self-supervised 2D image SOT methods [10], [11], [12] commonly follow the paradigm of generating a pseudo label in the first frame and applying a "cycle training" strategy (i.e., training forward and backward) to provide supervision signals. These approaches build on the widely adopted Siamese networks originally developed for fully supervised 2D SOT [8], [9]. They assume the pseudo label in the first frame is reliable and focus on minimizing information loss during cycle training. For example, [10] leverages a memory queue to fuse search features across frames, while[11] introduces a region mask to guide gradient propagation for better feature selection.

Although it is common to extend 2D SOT methods to 3D point cloud tracking in fully supervised settings [26], [27], [20], [28], [29], doing so in a self-supervised manner is more challenging. As discussed earlier, the sparse and low-frequency nature of point cloud data makes appearance matching unreliable without human labels. Furthermore, the lack of intermediate supervision in cycle training introduces additional instability, especially in the early stages of training.

B. Self-supervised object discovery with point clouds

In our method, we train a proposal network to discover potential target positions for the SOT task. This is related to self-supervised object discovery [30], [31], [32], [33], [34], which aims to discover traffic objects from raw point clouds without human annotations. The core idea is to cluster the point clouds to generate pseudo bounding boxes, which are then used to train a detection network in a self-supervised manner. For example, Wu et al. [33] incorporate shape priors in box fitting, while other works[30], [32], [34] utilize scene flow to capture motion patterns, based on the assumption that points belonging to the same object exhibit similar motion (i.e., rigid body motion).

Some methods [31], [32] further adopt iterative training to generalize object discovery from moving to static objects based on shared appearance features, aiming to detect all objects in a scene. In contrast, our approach focuses specifically on enhancing discriminative features for moving targets, as SOT primarily concerns objects in motion. We emphasize learning semantic features that support the discovery of dynamic targets, rather than static ones.

C. Self-supervised pre-training with point clouds

Self-supervised pre-training serves as a strong prior to complement missing information or general patterns before fine-tuning on the main task, in both 2D and 3D domains. In 2D, contrastive learning [35], [36] and masked image modeling [37], [38] are well-established approaches. These techniques have been extended to 3D point clouds [39], [40], [41], where the focus is typically on object-level representation learning. However, such pre-training is less effective for our SOT scenario, where understanding the global motion of scene objects is crucial. To address this, we adopt point cloud forecasting [22], [23], [24], [25] as a pre-training proxy task. This approach captures temporal dynamics by predicting future frames, thereby encoding meaningful motion patterns. Additionally, ray-casting is commonly used in volume rendering tasks such as novel view synthesis [42]. Khurana et al. [24] employ ray-casting to render occupancy-based point clouds in a self-supervised fashion. Inspired by this, we incorporate ray-casting into our pre-training to jointly learn scene geometry with motion, which enhances the representation power with geometric shape priors of the feature extractor used in our SOT framework.

III. Method

A. Overview

The overall framework of our method is illustrated in Fig. 2. For training the proposal network, we first perform Local Motion-Guided Labeling, where point clouds are clustered based on their local motion vectors to generate pseudo labels. Simultaneously, we pre-train a feature extractor using point cloud forecasting as a proxy task, enabling the network to learn global motion patterns and geometric structures. To enhance the quality of pseudo labels, we introduce an iterative refining process. In each round, high-level semantic features are assigned back to the point cloud as prototypes, guiding improved clustering and training of the proposal network. During the testing phase, the trained proposal network generates potential target candidates in each frame. A predictive motion filter is then applied to track the target by estimating its state based on the object's previous dynamics.

To better convey the motivation and internal logic of our design, we first describe Local Motion-Guided Labeling in Sec.III-B, followed by Global Motion and Geometry Structure Learning in Sec.III-C. Finally, we detail the Iterative Semantic Prototype Refining in Sec. III-D.

B. Local Motion-Guided Labeling

We begin by generating training pseudo labels for the proposal network. Let the point cloud sequence captured by a LiDAR sensor in autonomous driving scenarios be denoted as

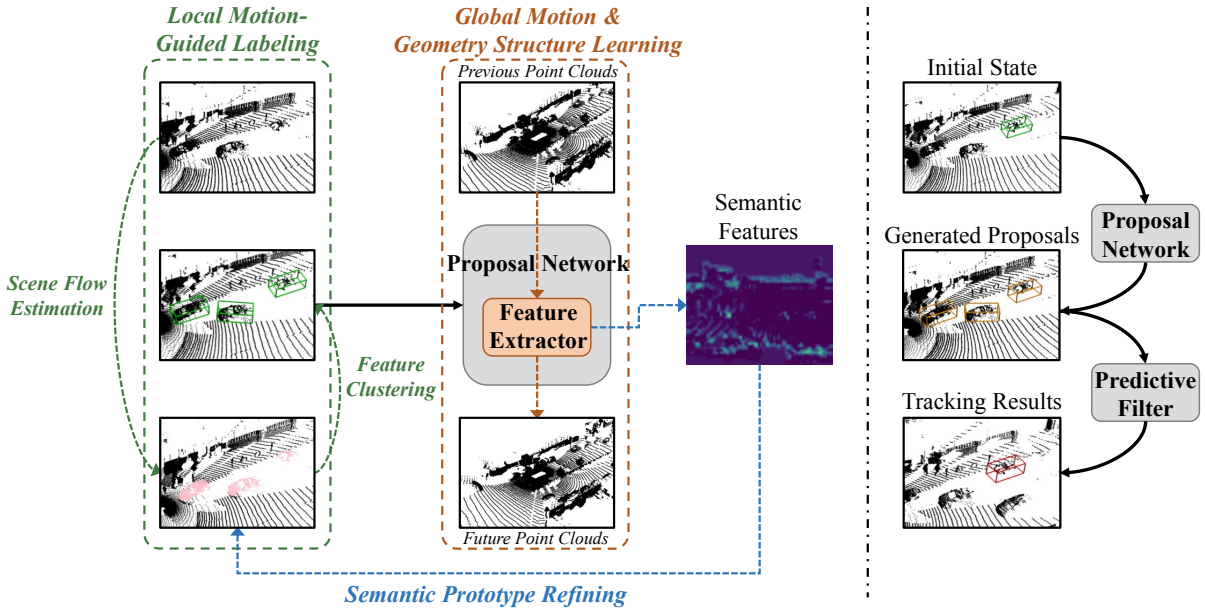

Fig. 2: Overview of our Self-Supervised Point Cloud Single Object Tracking (SPSOT) framework. (a) For training the proposal network, we generate pseudo labels by clustering local motion vectors, pre-train the feature extractor to learn global motion and geometry structures, and iteratively refine clustering with semantic prototypes. (b) To obtain tracking results, the trained proposal network generates candidates, and a predictive filter tracks the target.

$\mathbf{P}_{1:T} = \{\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_T\} \in \mathbb{R}^{T\times N\times 3}$, where $\mathbf{P}_t$ represents the point cloud at frame t, and N is the number of points in each frame. Each point cloud $\mathbf{P}_t$ can be partitioned into moving points $\mathbf{P}_t^{\mathrm{m}}$ and static points $\mathbf{P}_t^{\mathrm{s}}$. Although the target may occasionally be present in $\mathbf{P}_t^{\mathrm{s}}$ (i.e., when it is temporarily stationary), identifying it among static background points is extremely challenging without annotations. Therefore, we treat $\mathbf{P}_t^{\mathrm{s}}$ as background and focus solely on the moving points $\mathbf{P}_t^{\mathrm{m}}$. We argue that these occasional stationary frames do not significantly hinder the learning of target representations, as the majority of frames exhibit target motion. To identify $\mathbf{P}_t^{\mathrm{m}}$, we estimate scene flow vectors between consecutive frames.

Given two successive point clouds $\mathbf{P}_t, \mathbf{P}_{t+1} \in \mathbb{R}^{N\times 3}$, our goal is to estimate a scene flow $\mathbf{M}_{t\to t+1} \in \mathbb{R}^{N\times 3}$, where each vector describes the 3D motion of a point in $\mathbf{P}_t$ toward its expected position in $\mathbf{P}_{t+1}$. This flow estimation can be performed using any self-supervised scene flow method; in our case, we adopt SLIM [43]:

$$\mathbf{M}_{t\to t+1} = \mathrm{FlowEstimation}(\mathbf{P}_t, \mathbf{P}_{t+1}). \tag{1}$$

However, scene flow vectors $\mathbf{M}_{t\to t+1}$ inherently entangle ego-motion $\mathbf{M}_{t\to t+1}^{\mathrm{ego}}$ (i.e., the movement of the LiDAR platform itself) and the motion of external objects $\mathbf{M}_{t\to t+1}^{\mathrm{obj}}$, making it difficult to isolate the true object motions. Fortunately, open-source autonomous driving datasets often provide high-precision ego-pose information via GNSS (RTK GPS), IMU, and wheel encoders — data that does not rely on human annotation. Using this, we decouple ego-motion from scene flow. Assuming the ego-motion between frames t and $t+1$ is defined by a rotation matrix $\mathbf{R} \in \mathbb{SO}(3)$ and a translation vector $\mathbf{t} \in \mathbb{R}^3$, we compute the ego-motion and object-motion components as follows:

$$\begin{aligned} \mathbf{M}_{t\to t+1}^{\mathrm{ego}} &= \mathbf{R}\cdot\mathbf{P}_t + \mathbf{t} - \mathbf{P}_t, \\ \mathbf{M}_{t\to t+1}^{\mathrm{obj}} &= \mathbf{M}_{t\to t+1} - \mathbf{M}_{t\to t+1}^{\mathrm{ego}}. \end{aligned} \tag{2}$$

With the object motion vectors $\mathbf{M}_{t\to t+1}^{\mathrm{obj}}$ computed, we concatenate each point's physical position (x, y, z) with its motion vector $(\Delta x, \Delta y, \Delta z)$ to form a 6D feature. We then apply DBSCAN [44] to cluster these features and generate initial pseudo labels $\tilde{\mathcal{G}}$:

$$\tilde{\mathcal{G}} = \mathrm{DBSCAN}([\mathbf{P}_t, \mathbf{M}_{t\to t+1}^{\mathrm{obj}}]). \tag{3}$$

It can be observed that, within each pseudo label, the clustered points tend to share similar motion patterns within local physical neighborhoods. These pseudo labels highlight all moving objects in the scene and can serve as potential proposals for the target. To eliminate unreasonable pseudo labels, we follow the refinement strategy in [32], which includes filtering out bounding boxes with implausible shapes or abrupt appearances.

C. Global Motion and Geometry Structure Learning

As discussed earlier, the initial pseudo labels generated via scene flow rely on local motion cues. However, once the clustering is completed, this motion information is no longer

retained. Since single object tracking (SOT) fundamentally depends on understanding temporal continuity and motion, we aim to embed such awareness into the feature extractor of the proposal network. To this end, we pre-train the feature extractor to learn both global motion dynamics and geometric structures, enhancing its ability to capture spatiotemporal context.

We choose point cloud forecasting as a proxy task for this object mining process. Given a sequence of point clouds $\{\mathbf{P}_1, ..., \mathbf{P}_{t-1}, \mathbf{P}_t, \mathbf{P}_{t+1}, ..., \mathbf{P}_T\}$, the task involves predicting future frames $\mathbf{P}_{\text{fut}} = \{\mathbf{P}_{t+1}, ..., \mathbf{P}_T\}$ from past observations $\mathbf{P}_{\text{prev}} = \{\mathbf{P}_1, ..., \mathbf{P}_{t-1}, \mathbf{P}_t\}$. This setup allows the model to learn temporal motion features via an autoregressive encoding-decoding process. To simultaneously capture geometric structure, we avoid directly predicting future point coordinates. Instead, inspired by [24], we predict an occupancy probability field and reconstruct point clouds via differentiable ray-casting.

Let $\mathcal{F}_\theta$ denote the feature extractor (with parameters θ) and $\mathcal{H}_w$ the occupancy prediction head. We estimate occupancy probabilities over a voxelized grid $\mathbf{v}$ as:

$$\hat{\mathcal{O}}(\mathbf{v}) = \mathcal{H}_w(\mathcal{F}_\theta(\mathbf{P}_{\text{prev}})), \tag{4}$$

where $\hat{\mathcal{O}}(\mathbf{v}) : \mathbb{R}^3 \to [0, 1]$. Next, rays of the form $\mathbf{r}(s) = \mathbf{o} + s\mathbf{d},\ s \geq 0$, with origin $\mathbf{o} \in \mathbb{R}^3$ and direction $\mathbf{d} \in \mathbb{S}^2$, are cast through the occupancy field. The probability of the ray terminating at voxel $\mathbf{v}_i$ is computed as:

$$p_i = \prod_{j=1}^{i=1} (1 - \hat{\mathcal{O}}(\mathbf{v}_j)) \cdot \hat{\mathcal{O}}(\mathbf{v}_i), \tag{5}$$

where $\mathbf{v}_j$ are voxels traversed prior to reaching $\mathbf{v}_i$. Based on the stopping probabilities p_i, the expected distance $\hat{s}$ from the ray origin is defined as:

$$\hat{s} = \sum_{i=1}^{n} p_i \cdot s_i, \tag{6}$$

where n denotes the number of voxels along the ray. The future points $\mathbf{P}_{\text{fut}}$ are then reconstructed from the predicted distances $\hat{s}$ and the known directions.

Through this pre-training scheme, the feature extractor learns both global temporal motion (by forecasting future frames) and geometric structure (via occupancy-based ray rendering) for scene context understanding. These learned priors are critical for improving the tracking robustness in our self-supervised SOT framework.

D. Semantic Prototype Refining

To further refine the proposal network, we adopt an iterative training strategy that updates pseudo labels over successive rounds. Unlike previous self-supervised object discovery approaches [31], [32], which aim to gradually generalize toward static object discovery, our focus remains on improving the proposal network's ability to perceive moving targets. After each training round, we enhance the clustering process by assigning discriminative semantic features (referred to as prototypes) back to the original point cloud, thereby producing more accurate pseudo ground truths for the next iteration.

Specifically, given a point cloud $\mathbf{P} = \{\mathbf{p}_i = (x_i, y_i, z_i)\}_{i=1}^N$, we first project the 3D points into a 2D bird's-eye-view (BEV) grid using a discretization function $\pi : \mathbb{R}^3 \to \mathbb{Z}^2$:

$$\mathbf{g}_i = \pi(\mathbf{p}_i) = \left(\left\lfloor \frac{x_i}{r} \right\rfloor, \left\lfloor \frac{y_i}{r} \right\rfloor \right), \tag{7}$$

where $\mathbf{g}_i = (u_i, v_i)$ denotes the 2D grid coordinate of point $\mathbf{p}_i$, and r is the grid resolution. We adopt the PointPillars approach [45] for this discretization, under the assumption that points within the same pillar share similar semantic characteristics. Next, using the trained feature extractor $\mathcal{F}_\theta$ from the previous training iteration, we obtain a BEV semantic feature map:

$$\mathbf{F}_{\text{seman}} = \mathcal{F}_\theta(\mathbf{F}_{\text{BEV}}), \tag{8}$$

where $\mathbf{F}_{\text{BEV}}$ is the input BEV representation. To align with the original resolution of the BEV map, we upsample $\mathbf{F}_{\text{seman}}$ using bilinear interpolation to obtain $\mathbf{F}^{\text{up}}_{\text{seman}}$. Then, we retrieve the semantic feature for each point $\mathbf{p}_i$ via:

$$\mathbf{f}_i = \mathbf{F}^{\text{up}}_{\text{seman}}[:, \mathbf{g}_i], \tag{9}$$

where $\mathbf{f}_i$ represents the high-level semantic feature (i.e., prototype) associated with point $\mathbf{p}_i$. We combine these semantic prototypes $\mathbf{f}_i$ with each point's spatial coordinates and motion vectors to perform refined clustering:

$$\tilde{\mathcal{G}}' = \text{DBSCAN}([\mathbf{P}, \mathbf{M}^{\text{obj}}, \mathbf{F}_{\text{pts}}]), \tag{10}$$

where $\mathbf{F}_{\text{pts}} = \{\mathbf{f}_i\}_{i=1}^N$ denotes the full set of point-level semantic prototypes. This iterative refinement allows the clustering process to not only consider motion and positional information, but also reference semantic context, thereby reducing ambiguities in regions where multiple objects may exhibit similar motion patterns.

E. Predictive Motion Filter

Once the proposal network is trained, we integrate it with a predictive filter to produce the final tracking results. We use a Kalman filter [46] to model the temporal dynamics of the target. This provides an explicit motion constraint that propagates past state estimates into future frames, allowing us to identify the most probable target position from among the proposal candidates.

IV. Experiments

A. Implementation Details

All experiments are conducted using an NVIDIA RTX 4090 GPU and implemented with the PyTorch framework. For training the proposal network, we use an initial learning rate of 0.001 and apply a learning rate decay schedule. Training is performed for 30,000 steps per round over a total of three rounds. To avoid retaining noisy features from earlier stages, we reinitialize and retrain the proposal network from scratch at the start of each round. We apply non-maximum suppression (NMS) with a threshold of 0.1 to remove low-confidence proposals. For DBSCAN clustering, we use $\varepsilon = 1.0$ and

MinPts $= 5$. For scene flow estimation, we adopt the off-the-shelf SLIM [43] framework with its default settings. For point cloud forecasting, we pre-train our proposal network using the occupancy-based rendering head from [24], following its parameter configuration.

B. Datasets

Following common practices of supervised point cloud SOT methods, we evaluate our method on three popular autonomous driving datasets: KITTI [5], Waymo Open Dataset (WOD) [6], and nuScenes [7].

1) KITTI: KITTI is a foundational dataset that includes stereo cameras, a 64-beam Velodyne laser scanner and GPS/IMU systems. For this evaluation, we focus solely on the LiDAR data and use the KITTI Tracking dataset as [26], [28], [47], testing 2 sequences.

2) Waymo Open Dataset (WOD): The WOD is a large-scale dataset that spans both urban and suburban environments, providing a more complex set of challenges. It features 64-beam LiDAR data. Following [48], we evaluate our method across three difficulty levels: easy, medium, and hard, using 1121 tracklets.

3) nuScenes: The nuScenes dataset differs from the previous two in that it uses a 32-beam scanner, producing more sparse point cloud data, making it particulary challenging for testing on 150 scenes.

C. Evaluation metrics

We use the commonly adopted One-pass Evaluation (OPE) metrics in SOT tasks [49], including *Success* and *Precision*. *Success* measures the intersection over union (IoU) between the predicted 3D bounding box and the ground truth, while *Precision* calculates the area under the curve (AUC) for the distance between their centers (0–2 meters).

D. Baselines

Since our work is the first to tackle the fully self-supervised point cloud SOT task, there are no directly comparable methods. Therefore, we adapt existing self-supervised object discovery methods for static scenes, integrating a similar predictive filter as ours to achieve self-supervised point cloud SOT. Additionally, we include two supervised point cloud SOT methods for reference.

1) Unsupervised clustering method: DBSCAN [44] is widely used in point cloud spatial clustering. For this baseline, we directly apply DBSCAN to cluster the original points, forming sets of clusters. Then, we use a Kalman filter to predict and associate the nearest cluster as the prediction.

2) Self-supervised discovery methods: We choose two self-supervised discovery methods originally designed for static scenarios, combining them with the same Kalman filter for comparison. Oyster [31] first trains a detector on near-range point clusters and then retrains it on the full range. LISO [32] clusters both the point clouds' physical coordinates and motion vectors, using iterative training to generalize moving objects to static ones with similar appearances.

3) Supervised point cloud SOT methods: SC3D [29] is a supervised point cloud SOT method that leverages shape completion for tracking sparse appearance objects. P2B [28] is another supervised method we compare against, which integrates a 3D Region Proposal Network [50] to generate 3D proposals for tracking.

E. Comparisons

1) Results on KITTI: We compare our method with the baselines mentioned earlier on the KITTI dataset, as shown in Table I. Our method outperforms all baselines without requiring human annotations, which can be attributed to its global motion and geometry awareness, as well as its ability to discriminate moving objects. Notably, our method performs better than the supervised point cloud SOT method SC3D [29] in *Success* for the "Car" category, and outperforms P2B [28] in both *Success* and *Precision* for the "Cyclist" category. These successes are largely due to the prominent motion and easily identifiable features of these classes without human prompts. On the contrary, "Pedestrian" category lacks sufficient moving information, which often leads to confusion with other objects and noisy backgrounds. For further details, please refer to the limitations section (Sec. V).

TABLE I: Results on the KITTI dataset. SP, UN, SS represent supervised, unsupervised, and self-supervised methods, respectively.

Type	Method	Category Name					
		Success			Precision		
		Car	Pedestrian	Cyclist	Car	Pedestrian	Cyclist
SP	SC3D [29]	41.3	18.2	41.5	57.9	37.8	70.4
	P2B [28]	56.2	28.7	32.1	72.8	49.6	44.7
UN	DBSCAN [44]	8.2	0.2	5.6	10.1	0.1	2.1
SS	Oyster [31]	25.2	0.1	0.4	30.1	0.3	0.5
	LiSO [32]	38.4	10.1	38.9	44.3	**13.1**	40.2
	Ours	**42.1**	**10.2**	**40.4**	**49.8**	12.4	**44.8**

2) Results on WOD: Similar results are observed on the WOD dataset, as shown in Table II, where our method again outperforms all unsupervised and other self-supervised object discovery methods with a Kalman filter. The WOD dataset presents additional challenges due to its diverse scenes and varying object states, making it more difficult to achieve consistently high performance. Additionally, the "Vehicle" class includes a wide range of identities (e.g., cars, buses, trailers, motorcycles), complicating the identification. As in the KITTI results, the "Pedestrian" category remains a difficult tracking task, with the supervised method SC3D [29] showing only a slight improvement over our method. We also present qualitative results in Fig. 3, comparing our methods with LISO [32]. It is evident that LISO struggles to consistently track moving objects, which can lead to track drift.

3) Results on nuScenes: The nuScenes dataset is undoubtedly the most challenging of the three open datasets, primarily due to its 32-beam LiDAR data. As shown in Table III, our method achieves the highest *Success* and *Precision* without the use of human labels. Furthermore, it performs comparably

TABLE II: Results on the WOD dataset. The results are reported in *Success*/*Precision* format. SP, UN, SS represent supervised, unsupervised, and self-supervised methods, respectively.

Type	Method	Category Name					
		Vehicle			Pedestrain		
		Easy	Medium	Hard	Easy	Medium	Hard
SP	SC3D [29]	39.9/47.9	36.5/46.2	34.6/44.9	11.2/14.5	10.1/17.9	9.8/14.6
	P2B [28]	57.1/65.4	52.0/60.7	47.9/58.5	18.1/30.8	17.8/30.0	17.7/29.3
UN	DBSCAN [44]	6.5/7.3	4.5/5.1	3.4/3.6	3.1/1.0	2.5/0.8	1.4/0.8
SS	Oyster [31]	20.4/22.1	16.8/18.5	14.2/13.9	2.8/0.9	2.7/0.7	1.6/0.9
	LiSO [32]	27.5/ 30.8	21.7/ 23.7	22.7/ 24.5	8.0/12.2	6.8/9.3	7.1/10.6
	Ours	**32.6/ 36.7**	**25.1/ 26.7**	**25.4/ 26.9**	**8.5/13.5**	**7.1/10.4**	**8.8/10.9**

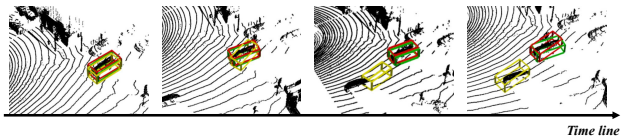

Fig. 3: Visualized comparison of our method with LISO [32] on WOD dataset. Green boxes denotes ground truths, red boxes are our predictions, and yellow boxes are predictions by LISO with a predictive Kalman filter.

to SC3D. Although categories like "Truck" and "Bus" share distinct motion characteristics with "Car", they suffer from severe class imbalance, making it harder for the proposal network to learn their representations.

TABLE III: Results on the nuScenes dataset. SP, UN, SS represent supervised, unsupervised, and self-supervised methods, respectively.

	Method	Category Name							
		Success				*Precision*			
		Car	Pedestrian	Truck	Bus	Car	Pedestrian	Truck	Bus
SP	SC3D [29]	22.3	11.3	30.7	29.4	21.9	12.7	27.7	24.1
	P2B [28]	38.8	28.4	43.0	33.0	43.2	52.2	41.6	40.1
UN	DBSCAN [44]	3.4	0.4	2.1	1.2	4.2	0.6	0.5	0.2
SS	Oyster [31]	14.6	3.2	8.6	9.7	14.2	2.6	8.4	8.8
	LiSO [32]	15.5	**7.1**	10.5	**12.9**	16.2	7.4	9.3	9.7
	Ours	**20.1**	6.3	**11.4**	12.8	**18.5**	**7.8**	**10.5**	**10.1**

F. Ablation studies

1) Motion and Geometry Learning: To verify the effectiveness of our Local Motion-Guided Labeling and Global Motion and Geometry Structure Learning, we conduct ablation studies shown in Table V. The results reveal that when initial labels are generated without motion vectors, performance drops significantly. Similarly, excluding the Global Motion and Geometry Structure Learning leads to noticeable decrease in performance. These findings highlight the importance of both components in improving the model's effectiveness. We further provide qualitative visualizations in Fig. 4, which illustrate the benefits of pre-training in enabling the proposal

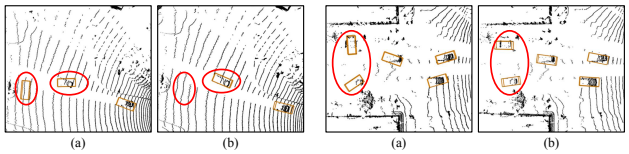

Fig. 4: Impact of Global Motion and Geometry Structure Learning on the WOD dataset. (a) Without pre-training. (b) With pre-training.

network to develop a more accurate understanding of object motion trajectories and structural shapes.

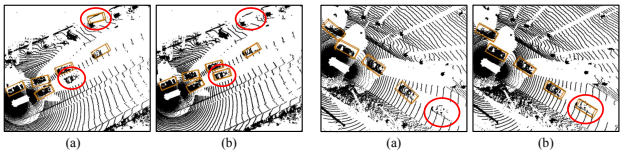

Fig. 5: Impact of Semantic Prototype Refining on the WOD dataset. (a) Without refining. (b) With refining.

2) Iterative Semantic Prototype Refining: The iterative training process and the refinement of the pseudo ground truth depends on two key factors: the number of training iterations and the feature dimensions engaged in the clustering process. We investigate these two factors on the KITTI "Car" category, as shown in Fig. 6. Since DBSCAN is sensitive to high dimension features [44], we experiment with different groupings of the 384-dimensional features, using 16, 32, and 64 groups, and apply average pooling to aggregate the semantic features in each group. Our results suggest that high-dimensional features may require careful tuning of hyperparameters to yield optimal performance. Based on our findings, we settle on 3 rounds of iterative training with 32-dimensional semantic features. As shown in Fig. 5, when semantic prototypes are used to guide iterative refinement, the proposal network learns more discriminative and robust representations of moving objects. This leads to improved identification of valid moving proposals and effective suppression of false positives caused by static background elements.

TABLE IV: Results of cycle training extension on the KITTI dataset. SP and SS denote supervised and self-supervised methods, respectively. P2B-CYC represents the cycle training extension of P2B.

Type	Method	Category Name					
		Success			Precision		
		Car	Pedestrian	Cyclist	Car	Pedestrian	Cyclist
SP	P2B [28]	56.2	28.7	32.1	72.8	49.6	44.7
SS	P2B-CYC	31.4	4.3	15.9	35.1	6.8	21.6
	Ours	**42.1**	**10.2**	**40.4**	**49.8**	**12.4**	**44.8**

3) Self-supervised Cycle Training Extension to 3D SOT: To investigate the feasibility of extending the "cycle training"

TABLE V: Ablation studies on the WOD "Vehicle" category. *w/o* MC&PT refers to the case where the scene motion vector is not used in clustering (only physical coordinates are considered), and global motion and geometry pre-training is excluded.

Method	WOD Vehicle		
	Easy	Medium	Hard
w/o MC&PT	24.2/28.1	17.4/19.2	16.9/19.8
w/o MC	26.8/31.2	21.8/23.3	22.7/23.0
w/o PT	28.4/33.9	22.7/24.5	24.2/25.4
Ours	**32.6/36.7**	**25.1/26.7**	**25.4/26.9**

strategy from 2D image-based SOT to 3D point cloud-based SOT, we adapt the original P2B [28] model with the cycle training approach proposed in [10]. The results are presented in Table IV. It can be observed that the Siamese-based cycle training method performs worse than our SPSOT framework, highlighting the challenge of appearance-based matching in the absence of human annotations.

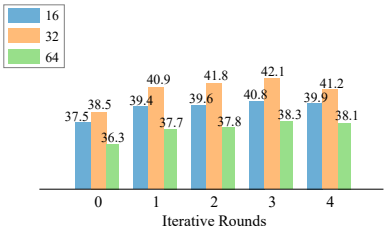

Fig. 6: Ablation studies on the KITTI "Car" category (reported in *Success*). X-axis represents the number of iterative training rounds, with three bars per round corresponding to different semantic feature dimensions.

V. Limitations

While we have achieved promising results with fully self-supervised point cloud SOT, there are two main limitations to our approach: 1) Since our method involves self-supervised training of a neural network, it is inevitably affected by long-tail data distribution problem. Specifically, the network tends to be biased towards the class with the most samples and the most accurate labels (e.g., cars), which can degrade performance on other classes. 2) Our method assumes that objects follow steady and distinct motion patterns. This limitation represents a common challenge in cases where supervision signals are weak, making it difficult to handle less distinctive categories such as pedestrians. Integrating RGB information could be a potential solution to address this issue.

VI. Conclusion

We proposed the first self-supervised point cloud SOT method integrating motion, geometry, and semantic information. To address the instability of appearance matching in Siamese networks, our framework generates proposals and then predicts the target based on its previous dynamics. We utilized scene flow for initial pseudo-label clustering and pre-train the proposal network to capture global motion and geometry patterns, and iteratively refine these labels using evolving semantic prototypes. Experiments on three datasets demonstrate performance comparable to fully supervised approaches.

Acknowledgment

This work was supported by the National Key R&D Program of China No. 2024YFC3015801, National Science Fund of China under Grant Nos. 62361166670, U24A20330, 62306238 and 62306155.

References

[1] Z. Zhu, Q. Meng, X. Wang, K. Wang, L. Yan, and J. Yang, "Curricular object manipulation in lidar-based object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 1125–1135.

[2] Y. Wang, K. Cheng, J. He, Q. Wang, H. Dai, Y. Chen, F. Xia, and Z.-X. Zhang, "Drivingdojo dataset: Advancing interactive and knowledge-enriched driving world model," *Proceedings of the Advances in Neural Information Processing Systems*, vol. 37, pp. 13 020–13 034, 2024.

[3] S. Zheng, W. Liu, Y. Guo, Y. Zang, S. Shen, C. Wen, M. Cheng, P. Zhong, and C. Wang, "Sr-adv: Salient region adversarial attacks on 3d point clouds for autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[4] W. Liu, W. Wang, Y. Qiao, Q. Guo, J. Zhu, P. Li, Z. Chen, H. Yang, Z. Li, L. Wang *et al.*, "Mmtl-uniad: A unified framework for multimodal and multi-task learning in assistive driving perception," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[5] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3354–3361.

[6] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2446–2454.

[7] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2020.

[8] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II 14*. Springer, 2016, pp. 850–865.

[9] A. He, C. Luo, X. Tian, and W. Zeng, "A twofold siamese network for real-time object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4834–4843.

[10] J. Zheng, C. Ma, H. Peng, and X. Yang, "Learning to track objects from unlabeled videos," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13 546–13 555.

[11] Q. Shen, L. Qiao, J. Guo, P. Li, X. Li, B. Li, W. Feng, W. Gan, W. Wu, and W. Ouyang, "Unsupervised learning of accurate siamese tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8101–8110.

[12] Z. Zhou, H. Fu, S. You, C.-C. J. Kuo *et al.*, "Uhp-sot++: An unsupervised lightweight single object tracker," *APSIPA Transactions on Signal and Information Processing*, vol. 11, no. 1, 2022.

[13] Y. Cui, Z. Fang, J. Shan, Z. Gu, and S. Zhou, "3d object tracking with transformer," in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2021, p. 317.

[14] C. Zheng, X. Yan, J. Gao, W. Zhao, W. Zhang, Z. Li, and S. Cui, "Box-aware feature enhancement for single object tracking on point clouds," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13 199–13 208.

[15] Z. Guo, Y. Mao, W. Zhou, M. Wang, and H. Li, "Cmt: Context-matching-guided transformer for 3d tracking in point clouds," in *Proceedings of the European Conference on Computer Vision*. Springer, 2022, pp. 95–111.

[16] J. Shan, S. Zhou, Z. Fang, and Y. Cui, "Ptt: Point-track-transformer module for 3d single object tracking in point clouds," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2021, pp. 1310–1316.

[17] C. Zhou, Z. Luo, Y. Luo, T. Liu, L. Pan, Z. Cai, H. Zhao, and S. Lu, "Pttr: Relational 3d point cloud object tracking with transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8531–8540.

[18] C. Zheng, X. Yan, H. Zhang, B. Wang, S. Cheng, S. Cui, and Z. Li, "Beyond 3d siamese tracking: A motion-centric paradigm for 3d single object tracking in point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8111–8120.

[19] Q. Wu, K. Sun, P. An, M. Salzmann, Y. Zhang, and J. Yang, "3d single-object tracking in point clouds with high temporal variation," in *Proceedings of the European Conference on Computer Vision*. Springer, 2025, pp. 279–296.

[20] T.-X. Xu, Y.-C. Guo, Y.-K. Lai, and S.-H. Zhang, "Cxtrack: Improving 3d point cloud tracking with contextual information," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 1084–1093.

[21] J. Liu, Y. Wu, M. Gong, Q. Miao, W. Ma, C. Xu, and C. Qin, "M3sot: Multi-frame, multi-field, multi-space 3d single object tracking," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, 2024, pp. 3630–3638.

[22] B. Mersch, X. Chen, J. Behley, and C. Stachniss, "Self-supervised point cloud prediction using 3d spatio-temporal convolutional networks," in *Conference on Robot Learning*. PMLR, 2022, pp. 1444–1454.

[23] X. Weng, J. Nan, K.-H. Lee, R. McAllister, A. Gaidon, N. Rhinehart, and K. M. Kitani, "S2net: Stochastic sequential pointcloud forecasting," in *Proceedings of the European Conference on Computer Vision*. Springer, 2022, pp. 549–564.

[24] T. Khurana, P. Hu, D. Held, and D. Ramanan, "Point cloud forecasting as a proxy for 4d occupancy forecasting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 1116–1124.

[25] Z. Yang, L. Chen, Y. Sun, and H. Li, "Visual point cloud forecasting enables scalable autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 14 673–14 684.

[26] L. Hui, L. Wang, L. Tang, K. Lan, J. Xie, and J. Yang, "3d siamese transformer network for single object tracking on point clouds," in *Proceedings of the European Conference on Computer Vision*. Springer, 2022, pp. 293–310.

[27] Z. Wang, Q. Xie, Y.-K. Lai, J. Wu, K. Long, and J. Wang, "Mlvsnet: Multi-level voting siamese network for 3d visual tracking," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3101–3110.

[28] H. Qi, C. Feng, Z. Cao, F. Zhao, and Y. Xiao, "P2b: Point-to-box network for 3d object tracking in point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6329–6338.

[29] S. Giancola, J. Zarzar, and B. Ghanem, "Leveraging shape completion for 3d siamese tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1359–1368.

[30] M. Najibi, J. Ji, Y. Zhou, C. R. Qi, X. Yan, S. Ettinger, and D. Anguelov, "Motion inspired unsupervised perception and prediction in autonomous driving," in *Proceedings of the European Conference on Computer Vision*. Springer, 2022, pp. 424–443.

[31] L. Zhang, A. J. Yang, Y. Xiong, S. Casas, B. Yang, M. Ren, and R. Urtasun, "Towards unsupervised object detection from lidar point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9317–9328.

[32] S. A. Baur, F. Moosmann, and A. Geiger, "Liso: Lidar-only self-supervised 3d object detection," in *Proceedings of the European Conference on Computer Vision*. Springer, 2024, pp. 253–270.

[33] H. Wu, S. Zhao, X. Huang, C. Wen, X. Li, and C. Wang, "Commonsense prototype for outdoor unsupervised 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 14 968–14 977.

[34] J. Seidenschwarz, A. Osep, F. Ferroni, S. Lucey, and L. Leal-Taixé, "Semoli: what moves together belongs together," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 14 685–14 694.

[35] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9729–9738.

[36] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the Inernational Conference on Machine Learning*. PmLR, 2020, pp. 1597–1607.

[37] K. He, X. Chen, S. Xie, Y. Li, P. Dollar, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2022.

[38] Z. Xie, Z. Zhang, Y. Cao, Y. Lin, J. Bao, Z. Yao, Q. Dai, and H. Hu, "Simmim: A simple framework for masked image modeling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2022.

[39] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu, "Point-bert: Pre-training 3d point cloud transformers with masked point modeling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 19 313–19 322.

[40] Y. Pang, W. Wang, F. E. Tay, W. Liu, Y. Tian, and L. Yuan, "Masked autoencoders for point cloud self-supervised learning," in *Proceedings of the European Conference on Computer Vision*. Springer, 2022, pp. 604–621.

[41] Q. He, J. Zhang, J. Peng, H. He, X. Li, Y. Wang, and C. Wang, "Pointrwkv: Efficient rwkv-like model for hierarchical point cloud learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, 2025, pp. 3410–3418.

[42] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[43] S. Andreas Baur, D. Josef Emmerichs, F. Moosmann, P. Pinggera, B. Ommer, and A. Geiger, "Slim: Self-supervised lidar scene flow and motion segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Oct 2021.

[44] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. AAAI Press, 1996, p. 226–231.

[45] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 697–12 705.

[46] T. Basar, *A New Approach to Linear Filtering and Prediction Problems*. Wiley-IEEE Press, 2001, pp. 167–179.

[47] K. Lan, H. Jiang, and J. Xie, "Temporal-aware siamese tracker: Integrate temporal context for 3d object tracking," in *Proceedings of the Asian Conference on Computer Vision*, 2022, pp. 399–414.

[48] Z. Pang, Z. Li, and N. Wang, "Model-free vehicle tracking and state estimation in point cloud sequences," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep 2021.

[49] M. Kristan, J. Matas, A. Leonardis, T. Vojir, R. Pflugfelder, G. Fernandez, G. Nebehay, F. Porikli, and L. Cehovin, "A novel performance evaluation methodology for single-target trackers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 2137–2155, Nov 2016.

[50] C. R. Qi, O. Litany, K. He, and L. Guibas, "Deep hough voting for 3d object detection in point clouds," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Oct 2019.