

# FlowTrack: Point-level Flow Network for 3D Single Object Tracking

Shuo Li, Yubo Cui, Zhiheng Li, and Zheng Fang\*

**Abstract**—3D single object tracking (SOT) is a crucial task in fields of mobile robotics and autonomous driving. Traditional motion-based approaches achieve target tracking by estimating the relative movement of target between two consecutive frames. However, they usually overlook local motion information of the target and fail to exploit historical frame information effectively. To overcome the above limitations, we propose a point-level flow method with multi-frame information for 3D SOT task, called FlowTrack. Specifically, by estimating the flow for each point in the target, our method could capture the local motion details of target, thereby improving the tracking performance. Meanwhile, to handle scenes with sparse points, we present a learnable target feature as the bridge to efficiently integrate target information from past frames. Moreover, we design a Instance Flow Head to transform dense point-level flow into instance-level motion, effectively aggregating local motion information to obtain global target motion. Finally, our method achieves competitive performance with improvements of 5.9% on the KITTI and 2.9% on the NuScenes, compared to the next best method.

## I. INTRODUCTION

3D single object tracking, an essential task in 3D computer vision, plays a crucial role in the fields of mobile robotics and autonomous driving. Given an initial target state, 3D SOT aims to capture the state of the target in subsequent frames, which can provide real-time information of target for downstream tasks like path planning and decision-making.

Inspired by 2D image tracking [1], [2], [3], the previous 3D SOT methods [4], [5], [6] usually treat object tracking as a matching problem. They utilize siamese networks to extract features from the template and the current frame, then rely on appearance matching to locate the target in the current frame. However, in scenes with object occlusion, interference from similar objects and distant target tracking, these matching-based methods face challenges due to the sparsity and disorder of point clouds.

Unlike the matching-based methods, some methods [7], [8] regard the tracking task as motion prediction and locate the target by predicting the relative motion between two adjacent frames. This motion-based paradigm alleviates the difficulty faced by appearance matching in the above challenge scenarios. However, these methods only focus on

This work was supported in part by the National Natural Science Foundation of China under Grants 62073066, in part by the Fundamental Research Funds for the Central Universities under Grant N2226001, and in part by 111 Project under Grant B16009. (Corresponding author: Zheng Fang.) The authors are all with the Faculty of Robot Science and Engineering, Northeastern University, Shenyang 110819, China. Shuo Li and Zheng Fang are also with the National Frontiers Science Center for Industrial Intelligence and Systems Optimization, Northeastern University, Shenyang 110819, China and also with the Key Laboratory of Data Analytics and Optimization for Smart Industry, Ministry of Education, Northeastern University, Shenyang 110819, China. (e-mail:fangzheng@mail.neu.edu.cn).

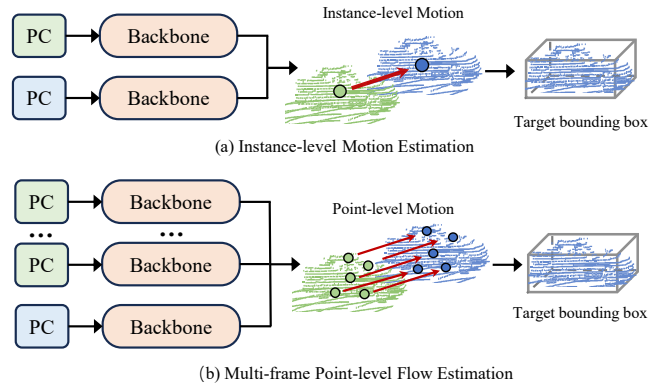


Fig. 1. Comparison of the current 3D single object tracking frameworks. (a) Instance-level motion estimation typically involves inputting two consecutive frames of point clouds and estimating the overall motion of the target. (b) Multi-frame point-level flow estimation involves inputting multiple historical frames of point clouds and estimating point-level flow within the target. Here, PC represents the input point clouds.

instance-level target motion and ignore the local motion details of the target. Additionally, they only utilize short-term motion cues between two frames, disregarding long-term geometry and motion information of the target from historical frames.

To address the above problems, we introduce the idea of optical flow estimation [9], [10] and propose a multi-frame point-level flow network, named **FlowTrack**. As shown in Fig. 1, unlike traditional motion-based methods, our proposed network can effectively capture the local motion details of the target by predicting the motion of each feature point in the target. Meanwhile, the rich geometric and motion information contained in historical frames offers valuable information supplements for estimating target motion, especially in scenes with sparse point clouds.

Specifically, FlowTrack first extracts BEV features of consecutive frame point clouds by 3D sparse convolution [11]. Then, to extract rich motion and geometry information of the target in historical frames, we propose a Historical Information Fusion Module (**HIM**) to efficiently fuse the information of target from historical frames into the template frame. Subsequently, we concatenate the template and current features along the channel as flow feature and use a proposed Point-level Motion Module (**PMM**) to extract the multi-scale point-level flow feature of target. Finally, with the aid of the Instance Flow head (**IFH**), our method accurately transforms the point-level flow of the target into instance-level motion, achieving precise relative motion estimation of the target. Extensive experiments have demonstrated that our method achieves superior tracking performance than existing

motion-based methods.

The main contributions of this paper are as follows:

- We propose a multi-frame point-level flow tracking framework to effectively capture target motion detail.
- We design HIM, PMM, and IFH to achieve historical information fusion, target point-level flow estimation, and instance-level motion prediction respectively.
- Our approach achieves competitive performance on the KITTI and nuScenes datasets. Extensive ablation studies demonstrate the effectiveness of the proposed modules.

## II. RELATED WORK

### A. 3D Single Object Tracking

Early 3D SOT methods [4], [5], [12] regard target tracking as a matching issue. They use the Siamese network to extract features of two consecutive frames and employ similarity matching for target localization. SC3D [4] is the first method to introduce the Siamese network structure into 3D SOT, enabling the capture of the target shape and subsequent generation of a series of bounding boxes within the search area. P2B [5] calculates point-level cosine similarity maps to enhance search features and adopts VoteNet for end-to-end prediction of the bounding box for target. PTTR [13] and STNet [6] utilize the Transformer-based structure [14] to promote intra-frame information exchange and inter-frame target association. STTracker [15] goes a step further by incorporating multi-frame information to improve the performance of matching approaches. Although the match-based paradigm has been successful in 3D SOT, it overlooks the motion information of the target. Additionally, the sparsity and occlusion of point clouds also pose significant challenges to appearance matching.

To alleviate this problem, M<sup>2</sup>-Track [7] proposes a motion-based approach, modeling the target’s relative motion instead of locating target through appearance-matching in two consecutive frames. DMT [8] designs a motion prediction module, which estimates the potential center of the target based on historical bounding boxes, and then refines it using a voting module. However, these methods only consider instance-level motion of target but overlook the local motion details of target. Besides, they do not effectively utilize the rich structural and motion information in historical frame.

### B. Optical Flow Estimation

Initially, the optical flow estimation is modeled as an optimization problem, aiming to maximize the visual similarity between image pairs with the regularizations [16], [17]. The advent of deep learning networks has significantly promoted advancements in this field. FlowNet [18] is the first end-to-end convolutional neural network for optical flow estimation, but it faces challenges in handling large object displacement motions. Afterward, RAFT [9] uses loop iteration at multiple scales to achieve excellent performance with the cost of increased processing time. GMFlow [10] further improves the accuracy and efficiency of optical flow estimation by combining similarity matching and global attention through

convolution networks. Furthermore, FlowFormer [19] represents the pioneering method that deeply integrates transformers with cost volumes for optical flow estimation. Inspired by the above methods, we integrate the concept of optical flow into the 3D SOT. By estimating the point-level flow of target, our network captures detailed local motion information of the target, thus achieving more accurate and robust tracking.

## III. METHODOLOGY

### A. Framework Overview

Given an initial bounding box  $\mathcal{B}_0$ , the 3D SOT task aims to estimate the  $\mathcal{B}_t = (x, y, z, w, h, l, \theta) \in \mathbb{R}^7$  in the subsequent frame. Here,  $(x, y, z)$ ,  $(w, l, h)$ , and  $\theta$  demote the 3D center, size and orientation, respectively. Moreover, the size of target is assumed to remain unchanged [5], thus we solely require to estimate the target center  $(x, y, z)$  and orientation  $\theta$ . For motion-based paradigm, given two consecutive frames  $\mathcal{P}_t$  and  $\mathcal{P}_{t-1}$  (termed as current and template frames) and target 3D bounding box  $\mathcal{B}_{t-1}$  in  $t-1$  frame, the tracker is required to predict the relative instance-level movement between the two adjacent frames, *i.e.* the translation offsets  $(\Delta x, \Delta y, \Delta z)$  and rotation angle  $\Delta\theta$ . Then, a rigid body transformation is applied to obtain the target bounding box  $\mathcal{B}_t$  in the current frame. This process could be expressed as follows:

$$\mathcal{F}(\mathcal{B}_{t-1}, \mathcal{P}_{t-1}, \mathcal{P}_t) \mapsto (\Delta x, \Delta y, \Delta z, \Delta\theta) \quad (1)$$

$$\mathcal{B}_t = \text{Transform}(\mathcal{B}_{t-1}, (\Delta x, \Delta y, \Delta z, \Delta\theta)) \quad (2)$$

Unlike the traditional motion-based approach, we replace the coarse relative instance-level motion by estimating the fine-grained **relative point-level flow** for the target, while incorporating historical frame information. Subsequently, through a proposed Instance Flow Head, we efficiently convert point-level flow into relative instance-level motion and derive  $\mathcal{B}_t$  for the current frame utilizing a rigid body transformation. Thus, our point-level flow paradigm could be formulated as follows:

$$\mathcal{F}(\mathcal{B}_{t-1}, \{\mathcal{P}_j\}_{j=t-T}^t) \mapsto \mathcal{M} \quad (3)$$

$$\mathcal{B}_t = \text{Transform}(\mathcal{B}_{t-1}, \text{IFH}(\mathcal{M})) \quad (4)$$

where  $\mathcal{M}$  is the relative point-level from  $t-1$  frame to the current  $t$  frame. And  $T$  represents the total number of frames of the input point cloud.

Based on Eq. 3 and Eq. 4, we propose a multi-frame point-level flow network, which performs SOT by estimating the relative point-level flow of the target. As displayed in Fig. 2, given the point clouds from historical and current frames, we first extract BEV features for each frame. Afterward, we design a Historical Information Fusion Module to efficiently integrate target features from historical features, which supplements the geometric and motion information of the target. Then, we concatenate the template and current features as flow feature and feed it into the Point-level Motion Module to capture local motion details of the target at multiple scales.

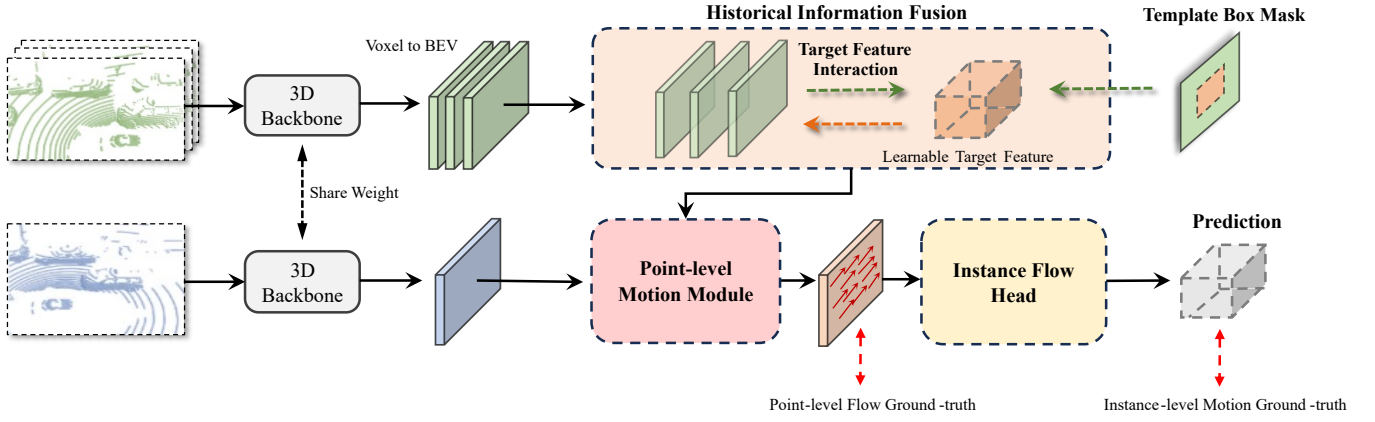


Fig. 2. **The overall framework of FlowTrack.** The network comprises a 3D Feature Extraction Backbone, Historical Information Fusion Module, Point-level Motion Module and Instance Flow Head. In each frame, the point cloud is initially processed by the 3D Feature Extraction Backbone to extract voxel features and transform them into BEV. Subsequently, the Historical Information Fusion Module effectively integrates information of the target from historical frames into the template frame. Then, the Point-level Motion Module is used to obtain multi-scale point-level flow features of the target. Finally, the Instance Flow Head adaptively transforms point-level flow into instance-level target motion to obtain the final results for target tracking.

Finally, the proposed Instance Flow Head predicts the point-level flow of the target and converts it into the instance-level relative motion of the target, achieving target tracking through rigid body transformation.

### B. Historical Information Fusion

Traditional motion-based algorithm [7] only utilizes two consecutive frames, overlooking the rich spatial and motion information of target in history frames. To extract target information in historical frames, we introduce a Historical Information Fusion Module based on a learnable target feature. It uses the learnable target feature as a bridge and employs Transformer [20] to achieve inter-frame target feature interaction, while reducing interference from background information in historical frames.

Following previous methods [21], we initially extract voxel features by voxelizing each frame’s point cloud and employing three layers of 3D sparse convolution. Then, we compress the height dimension of the voxel features from the final layer to generate multi-frame BEV features  $\{\mathcal{V}_k\}_{k=t-T}^t$ . Additionally, to highlight the target feature in the template feature, we define a template box mask  $\mathcal{Z}_{t-1} \in \mathbb{R}^{H \times W}$  by using the target box in template frame, where the mask value is set to 1 for grid points whose center lies within the box and 0 for others, as expressed in Eq. 5.

**Learnable Target Feature Initialization.** Based on the number of feature points occupied by the target in the template feature  $\mathcal{V}_{t-1}$ , we initially define a learnable target feature  $\mathcal{X} \in \mathbb{R}^{N \times C}$ . Here,  $N$  and  $C$  denote the number of feature points in the target and the channel. As shown in Fig. 3(a), we update it with the target feature in the template frame by cross-attention. Specifically, we first concatenate  $\mathcal{Z}_{t-1}$  with  $\mathcal{V}_{t-1}$  along the feature channels and pass it through a 2D convolution layer to obtain mask-enhanced  $\mathcal{V}_{t-1}'$ . Then, we use cross-attention to fuse the target information from  $\mathcal{V}_{t-1}'$  into  $\mathcal{X}$ , achieving the update of  $\mathcal{X}$ . By using  $\mathcal{Z}_{t-1}$  as guidance, we ensure  $\mathcal{X}$  learns target information rather than background information. This

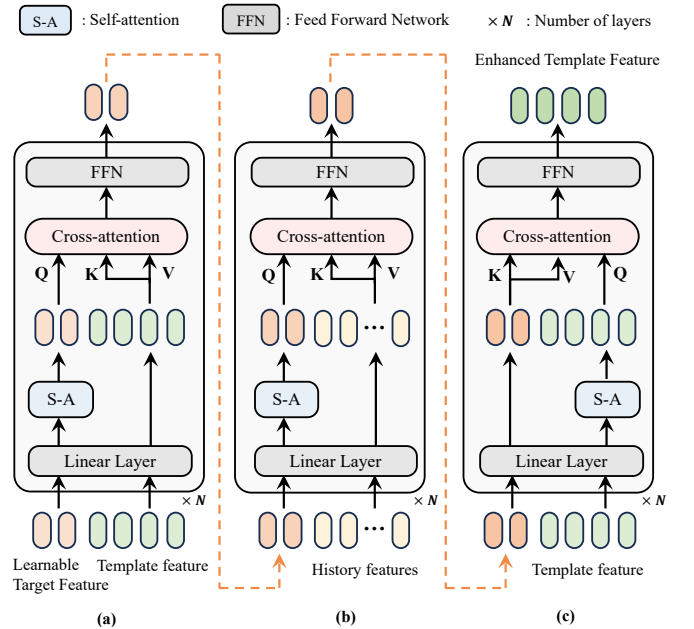


Fig. 3. **Details of Historical Information Fusion.** We use a learnable target feature as bridge to facilitate information interaction between historical frames and supplement the target feature in the template frame.

process can be expressed by the following formula:

$$\mathcal{Z}_{t-1} = \begin{cases} 1, & \text{if point in the box} \\ 0, & \text{others} \end{cases} \quad (5)$$

$$\mathcal{V}'_{t-1} = \text{Conv}(f(\mathcal{Z}_{t-1}, \mathcal{V}_{t-1})) \quad (6)$$

$$\mathcal{X}' = \text{CA}(\text{SA}(\text{Linear}(\mathcal{X})), \text{Linear}(\mathcal{V}'_{t-1}), \text{Linear}(\mathcal{V}'_{t-1})) \quad (7)$$

Here,  $\mathcal{X}'$  and  $f$  denote the updated learnable target feature and concatenation operation, while CA and SA stand for Cross-Attention and Self-Attention.

**Target Feature Interaction.** As shown in Fig. 3(b),(c), by using the updated  $\mathcal{X}'$  as bridge, the model can interact and complement the information of the target feature in the history frames, eventually enhancing the target feature in the

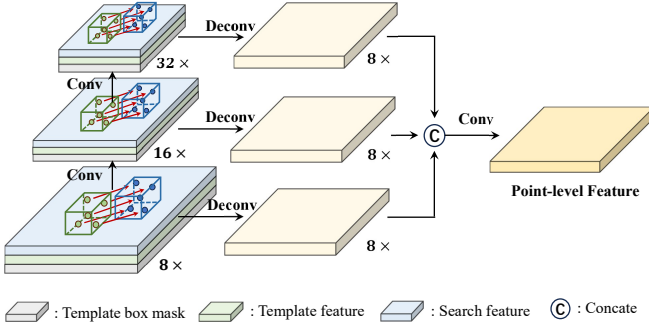


Fig. 4. **Details of Point-level Motion Module.** We utilize multiple layers of convolutional layers to extract multi-scale point-level flow motion features for tracking target. These features are then concatenated along the channel dimension to obtain the final point-level flow feature.

template frame. Similar to Eq. 7, we aggregate the target information from historical frames  $\{\mathcal{V}_j\}_{j=t-T}^{t-2}$  into  $\mathcal{X}'$  and then use the enhanced feature  $\mathcal{X}''$  to enrich the template feature  $\mathcal{V}_{t-1}$ . Specifically, using  $\mathcal{X}'$  as the query, history features  $\{\mathcal{V}_j\}_{j=t-T}^{t-2}$  as key and value, we get the enhanced target feature  $\mathcal{X}''$  by cross-attention, as shown in Fig. 3(b). Thus,  $\mathcal{X}''$  contains geometric and motion information of historical target. Finally, using  $\mathcal{V}_{t-1}$  as the query,  $\mathcal{X}''$  as the key and value, we integrate historical target information into the  $\mathcal{V}_{t-1}$  and get the enhanced template feature  $\mathcal{V}'_{t-1}$ , as shown in Fig. 3(c).

### C. Point-level Motion Module

Unlike motion-based methods that directly predict the instance-level motion of target, we capture local motion details within the target by predicting point-level flow, thereby achieving fine-grained motion estimation. However, different classes and sizes of tracked targets have different scales of motion. To fully capture their motion information, it is crucial to obtain point-level flow feature of the targets at multiple scales, as shown in Fig. 4.

To obtain the initial point-level flow feature  $F$ , we first concatenate the template box mask  $\mathcal{Z}_{t-1}$ , the enhanced template feature  $\mathcal{V}'_{t-1}$  and the current feature  $\mathcal{V}_t$  along the channel. Then, we use three layers of 2D convolution for downsampling to extract multi-scale point-level flow features  $\{F'_i\}_{i=0}^2$ , where  $i$  represents the label of the scale. Lastly, we upsample each feature in  $\{F'_i\}_{i=0}^2$  to the size of the input feature and concatenate them along the channel to obtain final point-level flow feature  $F''$ . The above process can be represented by the following procedure:

$$F = f(\mathcal{Z}_{t-1}, \mathcal{V}'_{t-1}, \mathcal{V}_t) \quad (8)$$

$$\{F'_i\}_{i=0}^2 = \mathcal{G}(F) \quad (9)$$

$$F'' = f(\text{Upsample}(\{F'_i\}_{i=0}^2)) \quad (10)$$

Here,  $\mathcal{G}$  represents a 3-layer convolution block, which performs three downsampling operations on the input  $F$ . In Eq. 9,  $\{F'_i\}_{i=0}^2$  is a collection of features from each convolutional layer output.

To train the Point-level Motion Module, we need to get point-level flow ground truth, denoted as  $P_r = \{p_{r(i,j)}\} (i \in$

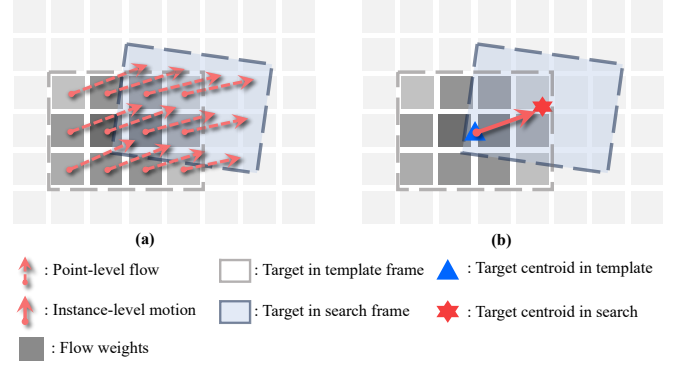


Fig. 5. **The transformation from point-level flow to instance-level motion.** (a) The point-level flow map, where grey squares represent the flow weight map; (b) The instance-level motion of the target.

$\{1, H\}, j \in \{1, W\}$ ). Initially, we project the point cloud and target bounding box of  $t$  frame to  $t-1$  frame to eliminate vehicle self-motion. Meanwhile, we define the center point position and orientation angle of target in these two frames as  $(C_{t-1}, C_t), (\theta_{t-1}, \theta_t)$ . Next, to obtain the ground truth  $P_r$ , we calculate the coordinate maps  $P_{t-1}, P_t$  in  $t-1$  and  $t$  frames, where  $P = \{p_{i,j}\}$  means the coordinates of each point on the feature map along the x-axis and y-axis.  $P_{t-1}$  can be easily obtained using the target bounding box  $\mathcal{B}_{t-1}$ , while  $P_t$  is computed by transforming  $P_{t-1}$  through translation and rotation, aiming to maintain the correspondence of target feature points across frames. Finally, we calculate  $P_r$  by subtracting the value of  $P_t$  and  $P_{t-1}$ . This process can be expressed by the following formula:

$$C_r = C_t - C_{t-1}, \theta_r = \theta_t - \theta_{t-1} \quad (11)$$

$$P_t = P_{t-1} \times \mathcal{T}(\theta_r) + C_r \quad (12)$$

$$P_r = P_t - P_{t-1} \quad (13)$$

Here,  $C_r$  and  $\theta_r$  represent the translation and rotation of the target, respectively.  $\mathcal{T}$  represents the transformation from orientation angle to rotation matrix.

### D. Instance Flow Head

To predict the point-level flow of the target and convert it into instance-level motion, we introduce the Instance Flow Head (IFH). It predicts the point-level flow map  $M_P \in \mathbb{R}^{H \times W \times 2}$  and the flow weight map  $W \in \mathbb{R}^{H \times W}$ . Here,  $M_P$  represents the relative motion of each point of the target on the feature map, while  $W$  denotes the weight of each point towards the overall motion. Specifically, using the input point-level flow feature  $F''$ , the IFH employs two separate convolutional layers to predict  $M_P$  and  $W$ . As shown in Fig. 5, the final instance-level target motion  $M_T$  is calculated by weighted averaging the value in  $M_P$  using  $W$ . The formula for this process is expressed as follows:

$$M_T = \frac{\sum_{i,j} M_P(i,j) \times W_{i,j}}{\sum_{i,j} W_{i,j}} \quad (14)$$

Here,  $i$  and  $j$  represent coordinates on the feature map.

Through the prediction of  $W$ , the network can dynamically learn the motion weights of various points within

TABLE I  
PERFORMANCE COMPARISON ON THE KITTI TEST DATASET.

	Method	Paradigm	Source	Car 6,424	Ped. 6,088	Van 1,248	Cyc. 308	Mean 14,068
Success / Precision	P2B [5]	Match	CVPR 2020	56.2/72.8	28.7/49.6	40.8/48.4	32.1/44.7	42.4/60.0
	PTT [22]		IROS 2021	67.8/81.8	44.9/72.0	43.6/52.5	37.2/47.3	55.1/74.2
	BAT [23]		ICCV 2021	60.5/77.7	42.1/70.1	52.4/67.0	33.7/45.4	51.2/72.8
	PTTR [13]		CVPR 2022	65.2/77.4	50.9/81.6	52.5/61.8	65.1/90.5	57.9/78.1
	SMAT [24]		RAL 2022	71.9/82.4	52.1/81.5	41.4/53.2	61.2/87.3	60.4/79.5
	STNet [6]		ECCV 2022	<u>72.1/84.0</u>	49.9/77.2	<b>58.0/70.6</b>	73.5/93.7	61.3/80.1
	GLT-T [25]		AAAI 2023	68.2/82.1	52.4/78.8	52.6/62.9	68.9/92.1	60.1/79.3
	CAT [26]		TNNLS 2023	66.6/81.8	51.6/77.7	53.1/69.8	67.0/90.1	58.9/79.1
	STTracker [15]		RAL 2023	66.5/79.9	60.4/89.4	50.5/63.6	<u>75.3/93.9</u>	62.6/82.9
	M <sup>2</sup> -Track [7]	Motion	CVPR 2022	65.5/80.8	61.5/88.2	53.8/ <b>70.7</b>	73.2/93.5	<u>62.9/83.4</u>
	DMT [8]		T-ITS 2023	66.4/79.4	48.1/77.9	53.3/65.6	70.4/92.6	55.1/75.8
	<b>Ours</b>		-	<b>74.8/84.4</b>	<b>65.0/92.5</b>	<u>54.3/65.3</u>	<b>79.1/94.7</b>	<b>68.8/86.4</b>
	Improvement	-	-	<u>↑2.7/↑0.4</u>	<u>↑3.5/↑3.1</u>	<u>↓3.7/↓5.4</u>	<u>↑3.8/↑0.8</u>	<u>↑5.9/↑3.0</u>

the target, effectively distinguishing between foreground and background points. Then, we obtain the position coordinates  $(x, y)$  of the target box along the x-axis and y-axis following Eq. 4. For the z-axis location coordinate and the orientation  $\theta$  of the target, we directly predict them by two independent convolution layers.

#### E. Training Loss

We adopt a similar approach to CenterPoint [27] for predicting the target’s relative motion along the x-axis and y-axis, z-axis position and orientation. Specifically, we predict the relative point-level flow map ( $M_P$ ), flow weight map ( $W$ ), z-axis location map ( $M_Z$ ) and orientation map ( $M_\theta$ ) with an  $8 \times$  downsampling. Following Eq. 14, the target’s relative motion  $M_T$  can be obtained from  $M_P$  and  $W$ . Additionally, we assume that points within the target box have the same z-axis position and orientation. Thus, the head loss can be defined as follows:

$$L_{head} = \lambda_{M_P} L_{M_P} + \lambda_{M_T} L_{M_T} + \lambda_{M_Z} L_{M_Z} + \lambda_{M_\theta} L_{M_\theta} \quad (15)$$

Here, we use L1 loss for  $L_{M_P}$ ,  $L_{M_T}$ ,  $L_{M_Z}$  and  $L_{M_\theta}$  for attribute regression.  $\lambda_{M_P}$ ,  $\lambda_{M_T}$ ,  $\lambda_{M_Z}$  and  $\lambda_{M_\theta}$  are the weights for these losses respectively. To balance the network’s consideration between point-level flow and flow weight map, we set the weight ratio of  $L_{M_P}$  to  $L_{M_T}$  as 1:2.

## IV. EXPERIMENTS

### A. Experimental Settings

**Datasets & Evaluation.** We train and evaluate our method on KITTI dataset [28] and nuScenes dataset [29]. KITTI dataset includes over 20,000 3D objects tracked in 21 scenes. nuScenes dataset comprises a total of 1,000 scenes, with each frame containing approximately 300,000 points. We follow the One Pass Evaluation (OPE) protocol [30] for model evaluation, which comprises two metrics: Success and Precision. Success quantifies the degree of overlap between predicted and ground truth bounding boxes, while Precision indicates the localization accuracy of the target.

TABLE II  
PERFORMANCE COMPARISON ON THE NuSCENES TEST DATASET. PED AND BIC IS AN ABBREVIATION FOR PEDESTRIAN AND BICYCLE.

	Method	Paradigm	Car 64,159	Ped. 33,227	Trailer 3,352	Bic. 2,292	Mean 103,030
Success	PTT [22]	Match	41.22	19.33	50.23	28.39	35.52
	P2B [5]		38.81	28.39	48.96	26.32	35.50
	BAT [23]		40.73	28.83	52.59	27.17	39.98
	SMAT [24]		43.51	32.27	37.45	25.74	39.29
	PTTR [22]		51.89	29.90	45.87	-	44.44
	STTracker [15]		<u>56.11</u>	<u>37.58</u>	48.13	36.23	<u>49.43</u>
	M <sup>2</sup> -Track [7]	Motion	55.85	32.10	<b>57.61</b>	<u>36.32</u>	47.81
	<b>Ours</b>		<b>60.29</b>	<b>37.60</b>	<u>55.39</u>	<b>43.74</b>	<b>52.35</b>
Improvement	-	<u>↑4.18</u>	<u>↑0.02</u>	<u>↓2.21</u>	<u>↑7.42</u>	<u>↑2.92</u>	
Precision	PTT [22]	Match	45.26	32.03	48.56	51.19	41.89
	P2B [5]		43.18	52.24	40.05	47.80	46.10
	BAT [23]		43.29	53.32	44.89	51.37	46.76
	SMAT [24]		49.04	60.28	34.10	61.06	52.45
	PTTR [22]		58.61	45.09	38.36	-	53.48
	STTracker [15]		<u>69.07</u>	<b>68.36</b>	55.40	<b>71.62</b>	<u>68.55</u>
	M <sup>2</sup> -Track [7]	Motion	65.09	60.92	<u>58.26</u>	67.50	66.77
	<b>Ours</b>		<b>71.07</b>	<u>67.64</u>	<b>62.70</b>	<u>71.43</u>	<b>69.70</b>
Improvement	-	<u>↑2.00</u>	<u>↓0.71</u>	<u>↑4.44</u>	<u>↓0.19</u>	<u>↑1.15</u>	

**Implementation Details.** We utilize sparse 3D convolution [21] widely used in 3D detection as our 3D feature extraction backbone. For our attention-based HIF Module, we follow common settings in Vision Transformer [31] with the number of heads set to 8 and the number of layers set to 6. During training, we employ the Adam optimizer with an initial learning rate of 0.0003 and weight decay of 0.01 for both datasets. Training consists of 80 epochs on KITTI and 40 epochs on nuScenes, utilizing 2 NVIDIA 4090 GPUs. We set a batch size of 64 per GPU during the training phase.

### B. Quantitative Results

**Results on KITTI.** As shown in Tab. I, our method outperforms the best matching-based paradigm, STTracker [15],

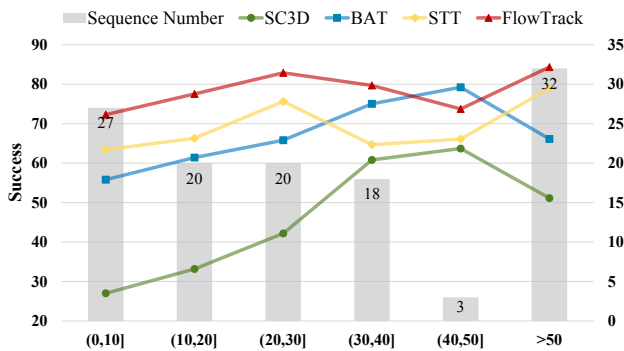


Fig. 6. Effect of the number of points for the target in the initial frame. In most cases, our FlowTrack significantly outperforms other methods.

by **6.2%/3.5%** in Success/Precision and surpasses the best motion-based paradigm, M<sup>2</sup>-Track [7], by **5.9%/3.0%** in Success/Precision. Specifically, Pedestrian targets typically present sparser point clouds and are more susceptible to interference from similar pedestrians. This poses challenges for appearance-based matching methods [6], [23], [13]. M<sup>2</sup>-Track [7] partially addresses this challenge by predicting instance-level relative motion for the tracking task. However, It overlooks the local motion details of the target and prior information from historical frames, resulting in suboptimal performance. Our method further transforms the tracking task into a point-level flow estimation, which effectively captures the local motion details of target, thereby improving the tracking performance in similar interference scenes. Meanwhile, in sparse tracking scenes, our method leverages the target information from multiple historical frames, resulting in a performance enhancement of **4.6%/3.1%** in Success/Precision for the pedestrian category.

**Results on nuScenes.** We validated our method on the larger tracking dataset nuScenes, and the result is shown in Tab. II. Our method surpasses the best motion-based method[7] by **4.54%/6.12%** in Success/Precision and outperforms the best match-based method [15] by **2.92%/1.15%** in Success/Precision. In particular, for the largest number of categories, vehicle, our method achieves outstanding performance improvements, outperforming the suboptimal method by **4.18%/2.00%** in Success/Precision. This is mainly attributed to the accurate capture of vehicle local motion details by the PMM module and the effective aggregation of historical information by the HIM module. Meanwhile, we observe that our model does not demonstrate superior performance when tracking large-size targets, such as Vans in KITTI and Trailers in nuScenes. We believe that the larger size of these targets results in significant variations in local point motion, thereby affecting the final instance-level motion estimation.

**Robustness to Sparseness.** In Fig. 6, consistent with prior studies [4], [23], [15], we display tracking performance under the different number of points for the target in the initial frame. our method surpasses alternative approaches in the majority of scenarios, specifically those with fewer than 40 points. This result confirms the distinct advantage of our method when dealing with sparse scenarios.

TABLE III  
COMPARISON OF THE RUNNING SPEEDS.

Method	PTT[22]	SMAT[24]	STNet[6]	M <sup>2</sup> -Track[7]
FPS	40.0	17.6	35.0	57.0
Method	GLT-T [25]	CAT [26]	STTracker [15]	Ours
FPS	30.0	45.3	23.6	33.2

TABLE IV  
ABLATION OF OUR COMPONENTS. HIM, PMM AND IFH RESPECTIVELY MEANS HISTORICAL INFORMATION FUSION, POINT-LEVEL MOTION MODULE AND INSTANCE FLOW HEAD

	HIM	PMM	IFH	3D Success	3D Precision
A1	✓	✓	✓	74.8	84.4
A2		✓	✓	71.2	80.6
A3	✓		✓	63.6	73.5
A4	✓	✓		71.4	81.6

**Running speed.** As shown in Tab. III, despite the introduction of multiple frames in our method, we still maintain a decent inference speed of 33.2 FPS for the model.

### C. Qualitative analysis

To better illustrate our method’s advantages in various scenes, we visualize some challenging scenes from the KITTI dataset, as shown in Fig. 7. In sparse and occluded scenes (the 1<sup>st</sup> and 2<sup>st</sup> rows), BAT, utilizing appearance matching, and M<sup>2</sup>-Track, relying on instance-level motion estimation, gradually lose track of the target. In contrast, our method utilizes the abundant target information from historical frames to provide prior knowledge for model tracking, which aids in recapturing the target once the point clouds become denser. In scenes with interference from similar objects (3<sup>st</sup> row), our point-level flow motion estimation captures finer local motions, thereby better distinguishing interfering objects and enhancing tracking performance.

### D. Ablation Study

To validate the effectiveness of our proposed modules, including Historical Information Fusion(HIM), Point-level Motion Module(PMM) and Instance Flow Head(IFH), we conduct ablation studies on the Car of the KITTI dataset.

**Model Components.** Firstly, we conduct an ablation study on the main components of our method to explore the effects of each module. As shown in Tab. IV, A1 represents all network models of our method. In A2, the removal of the HIM results in a performance decrease of 3.6%/3.8% in success/precision. This result demonstrates that the HIM effectively aggregates historical information, thereby improving the tracking performance of the network. In A3, when we remove the PMM, the model’s performance shows a significant decline, with success and precision decreasing by 11.2% and 10.9% respectively. This demonstrates the critical importance of multiscale point-level flow in target motion prediction, as it comprehensively captures the local motion details of the target. In A4, we replace the IFH with a traditional center head [27] and observe a decrease of 3.4%/3.3% in Success/Precision, respectively. It is evident that IFH

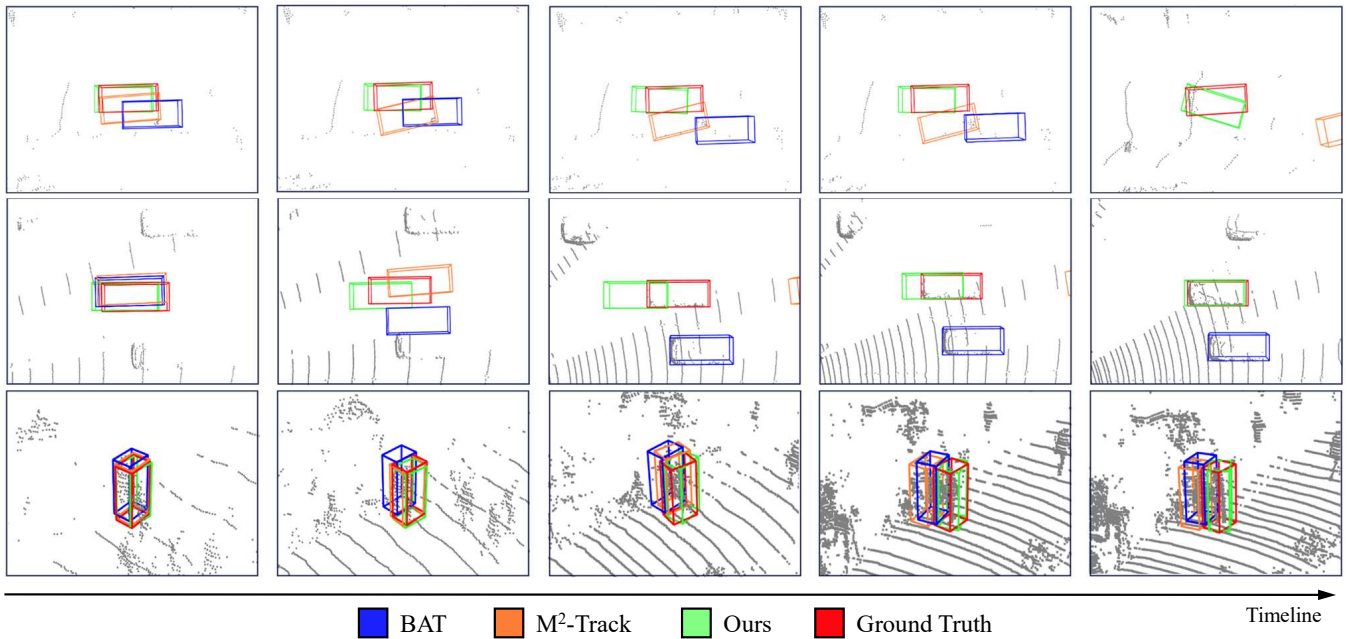


Fig. 7. **Visualization of tracking results on KITTI dataset.** From top to bottom, we compare our FlowTrack with BAT, M<sup>2</sup>-Track and show the cases of Car and Pedestrian categories. 1<sup>st</sup>: Point sparsity cases; 2<sup>st</sup>: Occlusion cases; 3<sup>st</sup>: Interference from similar targets.

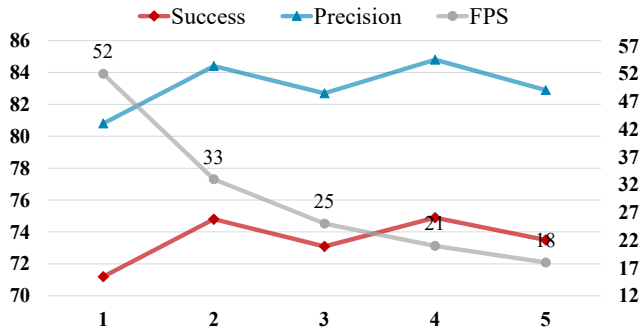


Fig. 8. **Performances with different number of historical frames.** We input different numbers of historical frames, from  $t-1$  to  $t-5$ . In the kitti dataset, the temporal frequency of the frames is 10 Hz. Our method could achieve higher performance with more previous frames.

accurately transforms dense point-level flow into instance-level motion, thereby enhancing tracking performance.

**Input History Frames.** To explore the impact of the number of historical frames  $\{\mathcal{P}_j\}_{j=t-T}^{t-1}$  on the model’s performance, we conducted an ablation experiment by adjusting the input historical frames. As illustrated in Fig. 8, our method showed significant enhancement as the historical frames increased from 1 to 2, improving performance by 3.6%/3.5%. However, as the historical frames increased further, performance fluctuated around 73.8%/83.5%. Earlier historical frames do not always contribute positively to the tracking of the target due to their inclusion of outdated target states. Additionally, an excessive number of historical frames can increase computational complexity and reduce the model’s speed. Hence, to balance speed and performance, we adopt the default setting of two history frames as input for our method, maintaining a performance of 33 FPS.

**Comparison in Historical Frames Fusion.** As shown in

TABLE V  
COMPARISON OF HISTORICAL FRAMES FUSION METHODS

	Methods	3D Success	3D Precision
B1	Concatenation	71.8	81.5
B2	Attention	72.1	81.7
B3	HIF	74.8	84.4

TABLE VI  
COMPARISON OF HEAD METHODS

	Methods	3D Success	3D Precision
C1	Center Head	71.4	81.1
C2	MLP	70.0	80.4
C3	Instance Flow Head	74.8	84.4

Tab. V, We conducted experimental comparisons on different historical frames fusion methods to validate the effectiveness of the HFM. In B1, we directly concatenate all historical frames along the feature channels. In B2, we employ a cross-attention mechanism, using  $\mathcal{V}_{t-1}$  as the query and  $\{\mathcal{V}_k\}_{k=t-T}^{t-2}$  as key and value. In B3, the HIF uses the learnable target feature as a bridge to achieve efficient information interaction between tracking targets across frames. The performance of the concatenation in B1 and the attention mechanism in B2 has decreased by 3.0%/2.9% and 2.7%/2.7% when compared to the HIF in B3. We believe that the operation of directly concatenating features from historical frames in B1 is overly simplistic for information aggregation, as it overlooks the alignment of target features across multiple frames. While the cross-attention in B2 achieves inter-frame information interaction, the feature interaction across the entire feature maps often introduces

interference from background information. By using the learnable target feature, the HIF can more effectively focus on the feature fusion of target information within the frame, significantly reducing background information interaction and achieving superior results.

**Comparison in Head Methods.** In the Tab. VI, we compared the impact of different prediction heads on the model's performance. In comparison to the IFH, adopting the Center Head, MLP led to performance drops of 3.4%/3.3% and 4.8%/4.0%, respectively. The prediction heads in C1 and C2 do not effectively consider the transformation relationship from dense point-level flow motion to instance-level target motion, thereby leading to a decline in performance. In contrast, our IFH achieves accurate transformation between the two by adaptively predicting flow weight maps.

## V. CONCLUSION

In this paper, we propose a multi-frame point-level flow tracking network, FlowTrack, which achieves target tracking by capturing the local motion information of the target. Simultaneously, the proposed HIF and IFH effectively aggregate target information across temporal dimensions and convert the point-level flow of target to instance-level motion. Extensive experiments demonstrate that our method outperforms previous matching and motion paradigm methods. However, our method does not exhibit superior performance when tracking large-sized objects, likely due to the fewer central BEV features. In future work, we plan to incorporate sparse point features to predict point-level flow, aiming to achieve better performance.

## REFERENCES

- [1] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," in *ECCV Workshops* (G. Hua and H. Jégou, eds.), pp. 850–865, 2016.
- [2] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *IEEE/CVF CVPR*, pp. 8971–8980, 2018.
- [3] D. Guo, J. Wang, Y. Cui, Z. Wang, and S. Chen, "Siamcar: Siamese fully convolutional classification and regression for visual tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6269–6277, 2020.
- [4] 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.
- [5] H. Qi, C. Feng, Z. Cao, F. Zhao, and Y. Xiao, "P2b: Point-to-box network for 3d object tracking in point clouds," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.
- [6] L. Hui, L. Wang, L.-Y. Tang, K. Lan, J. Xie, and J. Yang, "3d siamese transformer network for single object tracking on point clouds," in *European Conference on Computer Vision*, 2022.
- [7] 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 *IEEE CVPR*, pp. 8111–8120, 2022.
- [8] Y. Xia, Q. Wu, W. Li, A. B. Chan, and U. Stilla, "A lightweight and detector-free 3d single object tracker on point clouds," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 5, pp. 5543–5554, 2023.
- [9] Z. Teed and J. Deng, "Raft: Recurrent all-pairs field transforms for optical flow," in *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pp. 402–419, Springer, 2020.
- [10] A. Luo, F. Yang, X. Li, L. Nie, C. Lin, H. Fan, and S. Liu, "Gafflow: Incorporating gaussian attention into optical flow," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9642–9651, 2023.
- [11] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," in *Sensors*, vol. 18, 2018.
- [12] L. Hui, L. Wang, M. Cheng, J. Xie, and J. Yang, "3d siamese voxel-to-bev tracker for sparse point clouds," in *Neural Information Processing Systems*, 2021.
- [13] C. Zhou, Z. Luo, Y. Luo, T. Liu, L. Pan, Z. Cai, H. Zhao, and S. Lu, "Ptrr: Relational 3d point cloud object tracking with transformer," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8521–8530, 2021.
- [14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., 2017.
- [15] Y. Cui, Z. Li, and Z. Fang, "Sstracker: Spatio-temporal tracker for 3d single object tracking," *IEEE Robotics and Automation Letters*, vol. 8, pp. 4967–4974, 2023.
- [16] M. J. Black and P. Anandan, "A framework for the robust estimation of optical flow," in *1993 (4th) International Conference on Computer Vision*, pp. 231–236, IEEE, 1993.
- [17] A. Bruhn, J. Weickert, and C. Schnörr, "Lucas/kanade meets horn/schunck: Combining local and global optic flow methods," *International journal of computer vision*, vol. 61, pp. 211–231, 2005.
- [18] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, pp. 2758–2766, 2015.
- [19] Z. Huang, X. Shi, C. Zhang, Q. Wang, K. C. Cheung, H. Qin, J. Dai, and H. Li, "Flowformer: A transformer architecture for optical flow," in *European conference on computer vision*, pp. 668–685, Springer, 2022.
- [20] T. Meinhardt, A. Kirillov, L. Leal-Taixé, and C. Feichtenhofer, "Trackformer: Multi-object tracking with transformers," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8834–8844, 2021.
- [21] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4490–4499, 2017.
- [22] J. Shan, S. Zhou, Z. Fang, and Y. Cui, "Ptt: Point-track-transformer module for 3d single object tracking in point clouds," *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1310–1316, 2021.
- [23] 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," *2021 IEEE/CVF International Conference on Computer Vision*, pp. 13179–13188, 2021.
- [24] Y. Cui, J. Shan, Z. Gu, Z. Li, and Z. Fang, "Exploiting more information in sparse point cloud for 3d single object tracking," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11926–11933, 2022.
- [25] J. Nie, Z. He, Y. Yang, M. Gao, and J. Zhang, "Glt-t: Global-local transformer voting for 3d single object tracking in point clouds," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, pp. 1957–1965, 2023.
- [26] J. Gao, X. Yan, W. Zhao, Z. Lyu, Y. Liao, and C. Zheng, "Spatio-temporal contextual learning for single object tracking on point clouds," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2023.
- [27] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11784–11793, 2021.
- [28] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, 2012.
- [29] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nusenes: A multimodal dataset for autonomous driving," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.
- [30] M. Kristan, J. Matas, A. Leonardis, T. Vojtíř, R. Pflugfelder, G. Fernández, G. Nebehay, F. Porikli, and L. Čehovin, "A novel performance evaluation methodology for single-target trackers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 11, pp. 2137–2155, 2016.
- [31] S. Gao, C. Zhou, C. Ma, X. Wang, and J. Yuan, "Aiatrack: Attention in attention for transformer visual tracking," in *European Conference on Computer Vision*, pp. 146–164, Springer, 2022.