3D Siamese Transformer Network for Single Object Tracking on Point Clouds

Le Hui, Lingpeng Wang, Linghua Tang, Kaihao Lan, Jin Xie*, and Jian Yang*

Key Lab of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education
Jiangsu Key Lab of Image and Video Understanding for Social Security
PCA Lab, School of Computer Science and Engineering
Nanjing University of Science and Technology, China
{le.hui,cslpwang,tanglinghua,lkh,csjxie,csjyang}@njust.edu.cn

Abstract. Siamese network based trackers formulate 3D single object tracking as cross-correlation learning between point features of a template and a search area. Due to the large appearance variation between the template and search area during tracking, how to learn the robust cross correlation between them for identifying the potential target in the search area is still a challenging problem. In this paper, we explicitly use Transformer to form a 3D Siamese Transformer network for learning robust cross correlation between the template and the search area of point clouds. Specifically, we develop a Siamese point Transformer network to learn shape context information of the target. Its encoder uses self-attention to capture non-local information of point clouds to characterize the shape information of the object, and the decoder utilizes cross-attention to upsample discriminative point features. After that, we develop an iterative coarse-to-fine correlation network to learn the robust cross correlation between the template and the search area. It formulates the cross-feature augmentation to associate the template with the potential target in the search area via cross attention. To further enhance the potential target, it employs the ego-feature augmentation that applies self-attention to the local k-NN graph of the feature space to aggregate target features. Experiments on the KITTI, nuScenes, and Waymo datasets show that our method achieves state-of-the-art performance on the 3D single object tracking task. Source code is available at https://github.com/fpthink/STNet.

Keywords: 3D Single Object Tracking, Siamese Network, Transformer, Point Clouds

1 Introduction

Object tracking is a classic task in computer vision, and contributes to various applications, such as autonomous driving [39,41,34], visual surveillance [72,59],

* Corresponding authors.

robotics vision [38,8]. Early efforts [32,33,1,11,61] focus on visual object tracking that uses RGB images obtained by cameras. Recently, with the development of 3D sensors, such as LiDAR, 3D data is easy to acquire and set up 3D object tracking. Single object tracking is an important task in 3D computer vision. For example, it can improve the safety of autonomous vehicles by predicting the trajectory of key targets. However, due to the sparsity and irregular distribution of 3D points, existing popular schemes on 2D visual tracking cannot be directly applied to 3D single object tracking. Therefore, how to effectively track 3D objects in the complex scene is still a challenging problem.

Recently, Siamese network based trackers have raised much attention in the 3D single object tracking task. In [19], Giancola *et al.* first proposed a shape completion based 3D Siamese tracker, which encodes shape information into a template to improve the matching accuracy between the template and candidate proposals in the search area. However, it is time-consuming and not an end-to-end method. To this end, Qi *et al.* [50] proposed the point-to-box (P2B) network, which can be trained end-to-end and has a shorter inference time. Based on PointNet++ [49], P2B employs a target-specific feature augmentation module for the cross-correlation operation and adopts VoteNet [47] to regress the target center from the search area. Based on P2B, zheng *et al.* [78] proposed a box-aware tracker by inferring the size and the part priors of the target object from the template to capture the structure information of the target. Shan *et al.* [52] added a self-attention module in the VoteNet. Due to sparse point clouds, VoteNet is not suitable for regressing the target center in outdoor scenes. Lately, based on the bird's eye view feature map, Cui *et al.* [9] used cross-attention to learn the 2D relationship between the template and search to localize the target. In addition, Hui *et al.* [26] proposed a voxel-to-BEV tracker, which regresses the target center from the dense BEV feature map after performing shape completion in the search area. Nonetheless, due to the large appearance variation between template and search area, these simple cross-correlation feature learning cannot effectively characterize the correlation between them well.

In this paper, we propose a novel 3D Siamese Transformer tracking framework, which explicitly uses Transformer to learn the robust cross correlation between the template and search area in 3D single object tracking. Specifically, we first develop a Siamese point Transformer network by learning long-range contextual information of point clouds to extract discriminative point features for the template and search area, respectively. Our Siamese point Transformer network is an encoder-decoder structure. On each layer of the encoder, after aggregating the local features of the point cloud, we develop a non-local feature embedding module, which uses self-attention to capture the non-local information of point clouds. It is desired that the points can utilize the non-local features from the same instance to capture the whole structure of the object, *i.e.*, shape information. In the decoder, we propose an adaptive feature interpolation module to propagate features from subsampled points to the original points to generate discriminative point features. Compared with the commonly used distance-based interpolation [49], the adaptive feature

interpolation can effectively obtain discriminative point features through the learnable weights of the attention. Once we obtain discriminative point features of the template and the search area, we further develop an iterative coarse-to-fine correlation network to learn the cross-correlation between them for localizing the target in the search area. It consists of a cross-feature augmentation module and an ego-feature augmentation module. In the cross-feature augmentation module, we fuse the two feature maps from the template and search area by building cross-correlation between them via cross-attention. In this way, the template information is embedded into the search area for localizing the potential target. Once we localize the potential target, we use ego-feature augmentation to further enhance the potential target by applying self-attention to the local k-NN graph in the feature space instead of using the common self-attention over the whole point clouds. By applying self-attention to the k-NN graph in the feature space, the point features with similar semantic information can be aggregated, thereby enhancing the potential target. By iteratively performing the cross-feature and ego-feature modules, we can obtain a more discriminative feature fusion map for identifying the target from the search area. Finally, we integrate the Siamese point Transformer network, the iterative cross-correlation network, and the detection network [26] to form the Siamese Transformer tracking framework. Experiments on the KITTI [18], nuScenes [4], and Waymo [57] datasets demonstrate the effectiveness of the proposed method on the 3D single object tracking task.

The contributions of this paper are as follows:

- We present a Siamese point Transformer network that explicitly uses the attention mechanism to form an encoder-decoder structure to learn the shape context information of the target.
- We develop an iterative coarse-to-fine correlation network that iteratively applies the attention mechanism to the template and the search area for learning robust cross-correlation between them.
- The proposed 3D Siamese Transformer network achieves state-of-the-art performance on the KITTI, nuScenes, and Waymo datasets in 3D single object tracking.

2 Related Work

3D single object tracking. Early single object tracking approaches [3,24] focus on 2D images. Recently, Siamese network based trackers [23,60,63,22,80] have significantly improved tracking performance compared to the traditional correlation filtering based trackers [25,13,12,75]. However, due to the lack of accurate depth information in RGB images, visual object tracking may not be able to accurately estimate the depth to the target.

Previous methods [56,40,46,2] adopt RGB-D data for 3D single object tracking. RGB-D based trackers [36,28,29] heavily rely on RGB information and adopt the same schemes used in visual object tracking with additional depth information. Although these approaches can produce very promising results, they

may fail when critical RGB information is degraded. Recently, researchers [19,45] have focused on using 3D point clouds for single object tracking. Giancola *et al.* [19] first proposed a shape completion based 3D Siamese tracker (SC3D) for single object tracking. It performs template matching between the template and plenty of candidate proposals generated by Kalman filter [20] in the search area, where a shape completion network is applied to the template for capturing the shape information of the object. Based on SC3D, Feng *et al.* [17] proposed a two-stage framework called Re-Track to re-track the lost objects of the coarse stage in the fine stage. However, SC3D cannot be end-to-end trained. To achieve end-to-end training, point-to-box (P2B) [50] first localizes the potential target center in the search area and then generates candidate proposals for verification. Due to incomplete targets in point clouds, box-aware tracker [78] based on P2B proposes a box-aware feature fusion module to embed the bounding box information given in the first frame to enhance the object features, where the size and part information of the template are encoded. Shan *et al.* [52] improved P2B by adding a self-attention module in the detector VoteNet [47] to generate refined attention features for increasing tracking accuracy. In addition, Cui *et al.* [9] proposed a Transformer-based method that first uses 3D sparse convolution to extract features to form a 2D BEV feature map and then uses Transformer to learn the 2D relationship between the template and search to localize the target. Lately, to handle sparse point clouds, Hui *et al.* [26] proposed a Siamese voxel-to-BEV tracker, which contains a Siamese shape-ware feature learning network and a voxel-to-BEV target localization network. It performs shape generation in the search area by generating a dense point cloud to capture the shape information of the target.

3D multi-object tracking. Different from 3D single object tracking, 3D multi-object tracking (MOT) usually adopts the tracking-by-detection paradigm [71,53,31,51]. 3D MOT trackers usually first use 3D detectors [54,55,47] to detect object instances for each frame and then associate the detected objects across all frames. Early 3D multi-object tracking approaches [7,68] adopt 3D Kalman filters to predict the state of associated trajectories and objects instances. In [68], Weng *et al.* first used PointRCNN [54] to obtain 3D detections from a LiDAR point cloud, and then combined 3D Kalman filter and Hungarian algorithm for state estimation and data association. With the wide adoption of deep neural networks, recent methods [76,70,65] use neural networks to learn the 3D appearance and motion features for increasing accuracy. Lately, Weng *et al.* [69] proposed a graph neural network that uses a graph neural network for feature interaction by simultaneously considering the 2D and 3D spaces. Yin *et al.* [74] first proposed CenterPoint to detect 3D objects on the point clouds and then used a greedy closest-point matching algorithm to associate objects frame by frame.

Transformer and attention. Transformer is first introduced in [62], which uses a self-attention mechanism [35] to capture long-range dependences of language sequences. Based on the Transformer, some further improvements have been proposed in various sequential tasks, including natural language

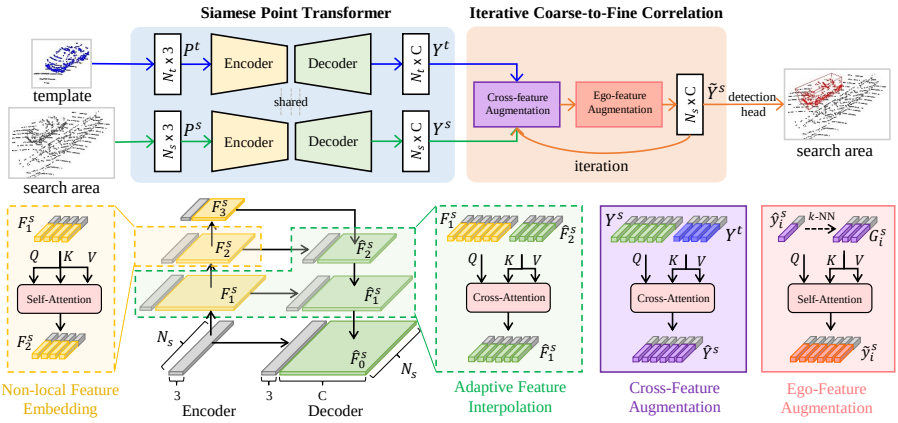

Fig. 1. The framework of our Siamese Transformer network. Given the template $\boldsymbol{P}^t$ and search area $\boldsymbol{P}^s$, we first use the Siamese point Transformer network to extract features $\boldsymbol{Y}^t$ and $\boldsymbol{Y}^s$ for the template and search area, respectively. Then, we perform the iterative coarse-to-fine correlation network to obtain a feature fusion map $\widetilde{\boldsymbol{Y}}^s$. Finally, we apply the detection head on the feature fusion map to localize the target. Note that Q, K, and V denote query, key, and value in Transformer, respectively.

processing [14,10,73], speech processing [58,42]. Recently, Dosovitskiy *et al.* [15] first proposed a vision Transformer for image recognition, which introduces a Transformer to handle non-sequential problems. The key idea is to split an image into patches, and feed the sequence of linear embeddings of these patches into a Transformer. After that, the Transformer is extended to various visual tasks, such as semantic segmentation [37,64], object detection [5,79], object tracking [6]. Recently, Liu *et al.* [37] proposed a hierarchical Transformer based on shifted windows to greatly reduce the computational cost while maintaining the capability to capture long-range dependencies in the data. For point cloud processing, Zhao *et al.* [77] first proposed a point Transformer that applies the self-attention mechanism on the local neighborhood of point clouds to extract local features for 3D semantic segmentation. Lately, inspired by point Transformer, different 3D vision tasks apply Transformer to yield good performance, such as point cloud classification [21], point cloud based place recognition [27], 3D object detection [44,43], 3D object tracking [9,52], and 3D action recognition [16].

3 Method

3.1 Siamese Point Transformer Network

In 3D single object tracking, given the target (*i.e.*, template) $\boldsymbol{P}^t = \{\boldsymbol{p}_i^t\}_{i=1}^{N_t}$ in the first frame, it aims to localize the 3D bounding box (3D BBox) of the same target in the search area $\boldsymbol{P}^s = \{\boldsymbol{p}_i^s\}_{i=1}^{N_s}$ frame by frame. N_t and N_s denote

the number of points in the template and search area, and $\boldsymbol{p}_i^t$ and $\boldsymbol{p}_i^s$ are 3D coordinates. With a slight abuse of notations we use the same symbols for the sets of points and for their corresponding matrices $\boldsymbol{P}^t \in \mathbb{R}^{N_t \times 3}$ and $\boldsymbol{P}^s \in \mathbb{R}^{N_s \times 3}$. The 3D BBox is formed as a 7-dimensional vector, which contains box center (x, y, z), box size (w, l, h), and yaw angle θ. Since the 3D BBox of the target is given in the first frame, we only need to regress the target center and yaw angle in the subsequent frames. By applying the displacement and yaw angle on the 3D BBox in the previous frame, the 3D BBox of the target in the current frame can be localized.

Most of existing Siamese trackers use local descriptors (such as PointNet [48] and PointNet++ [49]) as the feature extraction network. However, it lacks the ability to learn discriminative features by capturing long-range contextual information of point clouds. Thus, we propose a Siamese point Transformer network by utilizing attention to generate discriminative point features. As shown in Fig. 1, it is a hierarchical feature learning network, consisting of two key modules: non-local feature embedding and adaptive feature interpolation.

Non-local feature embedding. The encoder consists of three non-local feature embedding modules. The non-local feature embedding module executes self-attention on feature maps at different scales, capturing the contextual information at different scales of the point cloud, respectively. Given the search area $\boldsymbol{P}^s$ of N_s points, we follow P2B [50] to downsample the point cloud to generate point clouds at different scales by using random sampling. The number of the subsampled points in the l-th layer is $\frac{N_s}{2^l}$.

Specifically, in the l-th layer, we first execute the local feature embedding to capture local geometric structures of point clouds. Inspired by [66], we apply edge convolution on the k-nearest neighbors (k-NN) in the coordinate space to aggregate local features, denoted by $\boldsymbol{E}_l^s \in \mathbb{R}^{\frac{N_s}{2^l} \times C_l}$. Then, we perform the self-attention on the feature map $\boldsymbol{E}_l^s$ to learn long-range context information of the point cloud. Formally, the attention mechanism is defined as:

$$\boldsymbol{F}_l^s = \text{SelfAttention}(\boldsymbol{E}_l^s + \boldsymbol{X}_l^s, \boldsymbol{E}_l^s + \boldsymbol{X}_l^s, \boldsymbol{E}_l^s + \boldsymbol{X}_l^s) \tag{1}$$

where $\boldsymbol{X}_l^s \in \mathbb{R}^{\frac{N_s}{2^l} \times C_l}$ denotes the position embedding of the sampled points in the l-th layer. Note that position information of the point cloud is very important, and thus we add the positional embedding to all matrices. In Eq. (1), the three inputs from left to right are used as query, key, and value, respectively. The obtained feature map $\boldsymbol{F}_l^s$ in the l-th layer will be used as the input of the $(l+1)$-th layer. In this way, we can obtain feature maps $\boldsymbol{F}_1^s$, $\boldsymbol{F}_2^s$, and $\boldsymbol{F}_3^s$ at three scales.

Adaptive feature interpolation. After the encoder, the original point set is subsampled. As the number of points on the target is reduced, it is difficult to identify the target accurately. Although the distance based interpolation [49] can be used to interpolate new points, it cannot effectively interpolate high-quality point features for the target, especially in sparse point clouds. Thus, we design a learnable interpolation module to interpolate point features from the subsampled points to the original points through the learnable weights of the attention.

We define $\boldsymbol{F}_0 \in \mathbb{R}^{N_s \times 3}$ (*i.e.*, 3D coordinates) as the point feature of the original point with a size of N_s. Given the obtained feature maps $\boldsymbol{F}_1^s \in \mathbb{R}^{\frac{N_s}{2} \times C_1}$, $\boldsymbol{F}_2^s \in \mathbb{R}^{\frac{N_s}{4} \times C_2}$, $\boldsymbol{F}_3^s \in \mathbb{R}^{\frac{N_s}{8} \times C_3}$, and $\widehat{\boldsymbol{F}}_3^s = \boldsymbol{F}_3^s$, we gradually execute the adaptive feature interpolation to generate new point features from the low-resolution point cloud to the high-resolution point cloud, which is written as:

$$\widehat{\boldsymbol{F}}_l^s = \text{CrossAttention}(\boldsymbol{F}_l^s, \widehat{\boldsymbol{F}}_{l+1}^s, \widehat{\boldsymbol{F}}_{l+1}^s + \boldsymbol{X}_{l+1}^s) \tag{2}$$

where $l \in \{2, 1, 0\}$ and $\widehat{\boldsymbol{F}}_l^s \in \mathbb{R}^{\frac{N_s}{2^l} \times C_l}$ is the interpolated feature map. $\boldsymbol{X}_{l+1}^s \in \mathbb{R}^{\frac{N_s}{2^l} \times C_l}$ is the positional embedding. Note that the query $\boldsymbol{F}_l^s$ is the high-resolution feature map features, while the key ($\widehat{\boldsymbol{F}}_{l+1}^s$) and value ($\widehat{\boldsymbol{F}}_{l+1}^s + \boldsymbol{X}_{l+1}^s$) are the low-resolution feature maps. In Eq. (2), the feature map $\widehat{\boldsymbol{F}}_l^s$ is interpolated by weighting point features of the low-resolution (value) point cloud, considering the similarity between the high-resolution (query) and the low-resolution (key) point clouds. Finally, by applying the Siamese point Transformer on the template and search area, we obtain the feature maps $\widehat{\boldsymbol{F}}_0^t \in \mathbb{R}^{N_t \times C_0}$ and $\widehat{\boldsymbol{F}}_0^s \in \mathbb{R}^{N_s \times C_0}$ for the original point sets of the template and search area. For simplicity, we denote the obtained feature maps of the template and search area by $\boldsymbol{Y}^t \in \mathbb{R}^{N_t \times C}$ and $\boldsymbol{Y}^s \in \mathbb{R}^{N_s \times C}$. Note that $\boldsymbol{Y}^t = \widehat{\boldsymbol{F}}_0^t$ and $\boldsymbol{Y}^s = \widehat{\boldsymbol{F}}_0^s$.

3.2 Iterative Coarse-to-Fine Correlation Network

In 3D Siamese trackers, a cross-correlation operation is used to compute the similarity between the template and search area to generate a feature fusion map for identifying the target. Most of existing trackers use the cosine distance to generate the similarity map. Due to the large appearance variation between template and search area, this simple operation cannot effectively associate the template with the search area. Thus, we develop an iterative coarse-to-fine correlation network to learn the similarity in a coarse-to-fine manner to mitigate large appearance variation between them through the attention mechanism. Fig. 1 shows the detailed structure.

Cross-feature augmentation. We employ the cross-feature augmentation module to fuse the template and the search area by learning similarity between them. Given the template feature $\boldsymbol{Y}^t \in \mathbb{R}^{N_t \times C}$ and search area feature $\boldsymbol{Y}^s \in \mathbb{R}^{N_s \times C}$, we use the cross-attention mechanism between the template and search area to generate a coarse feature fusion map. Specifically, the cross-feature augmentation operation is formulated as:

$$\widehat{\boldsymbol{Y}}^s = \text{CrossAttention}(\boldsymbol{Y}^s, \boldsymbol{Y}^t, \boldsymbol{Y}^t + \boldsymbol{X}^t) \tag{3}$$

where $\widehat{\boldsymbol{Y}}^s \in \mathbb{R}^{N_s \times C}$ is the obtained coarse feature fusion map. Since the 3D coordinates of the template provide the positional relationship of the target, we add the positional embedding of the template $\boldsymbol{X}^t \in \mathbb{R}^{N_s \times C}$ to the value $\boldsymbol{Y}^t \in \mathbb{R}^{N_s \times C}$. By learning the similarity between the template (key) and search area (query), we embed the template (value) into the search area to generate

the feature fusion map $\widehat{\boldsymbol{Y}}^s$. In this way, the potential target in the feature fusion map can be associated with the template.

Ego-feature augmentation. Furthermore, we design an ego-feature augmentation module to enhance target information by considering the internal association in the coarse feature fusion map. Specifically, we first construct the k-nearest neighbor (k-NN) for each point in the feature space. Given the coarse feature fusion map $\widehat{\boldsymbol{Y}}^s \in \mathbb{R}^{N_s \times C}$ with N_s feature vectors $\widehat{\boldsymbol{y}}_i^s \in \mathbb{R}^C$, the similarity between points i and j is denoted as:

$$a_{i,j} = \exp(-\|\widehat{\boldsymbol{y}}_i^s - \widehat{\boldsymbol{y}}_j^s\|_2^2) \tag{4}$$

where $a_{i,j}$ is the similarity metric. For the point i, we select K nearest points in the feature space as its neighborhood by using the defined similarity. Thus, we obtain the local k-NN feature map for the point i, denoted by $\boldsymbol{G}_i^s \in \mathbb{R}^{K \times C}$. Since the points in the same instance have similar appearances, they are close to each other in the feature space. By aggregating local k-NN graphs in the feature space, the differences between different instances can be further magnified. Therefore, we then use the self-attention on the local k-NN graph to capture local association to aggregate discriminative point features. Specifically, given the point feature $\widehat{\boldsymbol{y}}_i^s \in \mathbb{R}^C$ and its k-NN feature map $\boldsymbol{G}_i^s \in \mathbb{R}^{K \times C}$, the local association is defined as:

$$\widetilde{\boldsymbol{y}}_i^s = \text{SelfAttention}(\widehat{\boldsymbol{y}}_i^s + \boldsymbol{x}_i^s, \boldsymbol{G}_i^s + \boldsymbol{Z}_i^s, \boldsymbol{G}_i^s + \boldsymbol{Z}_i^s) \tag{5}$$

where $\boldsymbol{x}_i^s \in \mathbb{R}^C$ and $\boldsymbol{Z}_i^s \in \mathbb{R}^{K \times C}$ are the positional embeddings of the i-th point and its k-NN neighborhood, and $\widetilde{\boldsymbol{y}}_i^s \in \mathbb{R}^C$ is the extracted point feature. In this way, we obtain the refined feature fusion map $\widetilde{\boldsymbol{Y}}^s \in \mathbb{R}^{N_s \times C}$.

We iteratively perform the coarse cross-feature augmentation module and the fine ego-feature augmentation module to generate a discriminative feature fusion map for identifying the target. Note that the output in the previous iteration will replace the search area input in the next iteration. By capturing the external (template) and internal (search area itself) relationships, our iterative coarse-to-fine correlation network can gradually generate a discriminative feature fusion map for identifying the target. Based on the feature fusion map, we use the 3D detector [26] to regress the target center and yaw angle.

4 Experiments

4.1 Experimental Settings

Datasets. We use the KITTI [18], nuScenes [4], and Waymo [57] datasets for single object tracking. For the KITTI dataset, it contains 21 video sequences. Following [19], we split the sequences into three parts: sequences 0-16 for training, 17-18 for validation, and 19-20 for testing. For the nuScenes dataset, it contains 700 sequences for training and 150 sequences for validation. Since the ground truth for the test set in nuScenes is inaccessible offline, we use its validation set to evaluate our method. For the Waymo dataset, we follow LiDAR-SOT [45] to use

Table 1. The performance of different methods on the KITTI and nuScenes datasets. Note that the results on the nuScenes dataset are obtained by using the pre-trained model on the KITTI dataset. "Mean" denotes the average results of four categories.

	Method	Success					Precision				
KITTI	Category	Car	Pedestrian	Van	Cyclist	Mean	Car	Pedestrian	Van	Cyclist	Mean
	Frame Num.	6424	6088	1248	308	14068	6424	6088	1248	308	14068
	SC3D [19]	41.3	18.2	40.4	41.5	31.2	57.9	37.8	47.0	70.4	48.5
	P2B [50]	56.2	28.7	40.8	32.1	42.4	72.8	49.6	48.4	44.7	60.0
	MLVSNet [67]	56.0	34.1	52.0	34.3	45.7	74.0	61.1	61.4	44.5	66.6
	LTTR [9]	65.0	33.2	35.8	66.2	48.7	77.1	56.8	45.6	89.9	65.8
	BAT [78]	60.5	42.1	52.4	33.7	51.2	77.7	70.1	67.0	45.4	72.8
	PTT [52]	67.8	44.9	43.6	37.2	55.1	81.8	72.0	52.5	47.3	74.2
	V2B [26]	70.5	48.3	50.1	40.8	58.4	81.3	73.5	58.0	49.7	75.2
	STNet (ours)	**72.1**	**49.9**	**58.0**	**73.5**	**61.3**	**84.0**	**77.2**	**70.6**	**93.7**	**80.1**
	Category	Car	Pedestrian	Truck	Bicycle	Mean	Car	Pedestrian	Truck	Bicycle	Mean
	Frame Num.	15578	8019	3710	501	27808	15578	8019	3710	501	27808
nuScenes	SC3D [19]	25.0	14.2	**25.7**	17.0	21.8	27.1	16.2	**21.9**	18.2	23.1
	P2B [50]	27.0	15.9	21.5	20.0	22.9	29.2	22.0	16.2	26.4	25.3
	BAT [78]	22.5	17.3	19.3	17.0	20.5	24.1	24.5	15.8	18.8	23.0
	V2B [26]	31.3	17.3	21.7	**22.2**	25.8	35.1	23.4	16.7	19.1	29.0
	STNet (ours)	**32.2**	**19.1**	22.3	21.2	**26.9**	**36.1**	**27.2**	16.8	**29.2**	**30.8**

1,121 tracklets, which are split into easy, medium, and hard subsets according to the number of points in the first frame of each tracklet. Following [26], we use the trained model on the KITTI dataset to test on the nuScenes and Waymo datasets for evaluating the generalization ability of our 3D tracker.

Evaluation metrics. We use *Success* and *Precision* defined in one pass evaluation [33] as the evaluation metrics for 3D single object tracking. Specifically, *Success* measures the intersection over union (IOU) between the predicted 3D bounding box (BBox) and ground truth (GT) box, while *Precision* measures the AUC for the distance between both two boxes' centers from 0 to 2 meters.

Network architecture. Following [50], we randomly sample $N_t = 512$ for each template P^t and $N_s = 1024$ for each search area P^s. For the Siamese point Transformer network, it consists of a three-layer encoder (three non-local feature embedding modules) and a three-layer decoder (three adaptive feature interpolation modules). In each encoder layer, the number of points is reduced by half. For example, if we feed the search area of 1024 points to the encoder, the number of points in each layer is 512, 256, and 128, respectively. Besides, the neighborhood sizes used to extract local feature are 32, 48, and 48, respectively. In the decoder, it gradually aggregates feature maps layer by layer to obtain a discriminative feature map. The obtained feature map sizes of the template $\boldsymbol{Y}^t$ and search area $\boldsymbol{Y}^s$ are 512×32 and 1024×32, respectively. For the iterative coarse-to-fine correlation network, we use two iterations considering the computational complexity and inference time. The hyperparameter K of the neighborhood size is set to 48. The size of the output feature fusion map $\widetilde{\boldsymbol{Y}}^s$ is 1024×32. Note that in this paper, we adopt the linear Transformer [30] and employ $n = 2$ attention heads for all experiments.

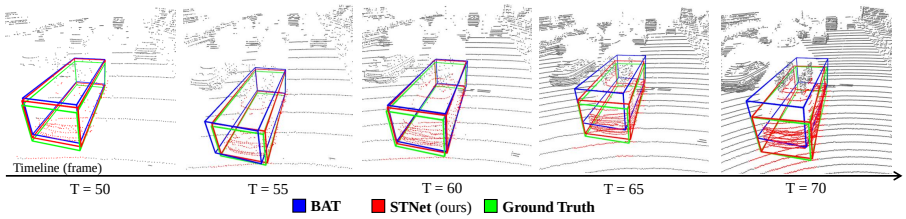

Fig. 2. The visualization results of our STNet and BAT on the car category of the KITTI dataset. The points on the target car are colored in red.

4.2 Results

Quantitative results. As shown in the top half of Tab. 1, we make comprehensive comparisons on the KITTI dataset with the previous state-of-the-art methods, including SC3D [19], P2B [50], MLVSNet [67], BAT [78], LTTR [9], PTT [52], and V2B [26]. Following [50], we report the results over four categories, including car, pedestrian, van, and cyclist. From the table, it can be found that our method outperforms other methods on the mean results of four categories. For the car category, our method can significantly improve the precision from 81.8% to 84.0% with a gain of about 2% on the car category. In addition to the large targets, our method can still achieve higher performance for those small targets, such as cyclists. For the cyclist category, compared with LTTR, our method obtains a gain of about 3% on the precision. In Fig. 2, we also show the visualization results of our method and BAT. It can be observed that our method (red boxes) can accurately localize the target. Most existing methods use local descriptors to extract point features. Due to the large appearance variation between template and search area, the extracted features cannot characterize the differences between them well. Thus, we propose a Siamese point Transformer network to learn the dense and discriminative point features with a learnable point interpolation module, where the shape context information of the target can be captured. Furthermore, we use an iterative coarse-to-fine correlation network to learn the similarity between the template and search area in a coarse-to-fine manner to mitigate large appearance variations in sparse point clouds for accurate object localization.

Visualization of attention maps. In Fig. 3, we show the attention maps generated by our method on the KITTI dataset, including car, pedestrian, van, and cyclist. The points marked with the red color can obtain high attentional weights. It can be observed that our method can accurately focus on the target in the search area. The visualization results show that when there are multiple objects, the target can be distinguished from the non-target objects. It means that the learned shape context information of the object can help to learn the discriminative relationship between the template and search area by Transformer.

Table 2. The performance of different methods on the Waymo dataset. Each category is split into three levels of difficulty: "Easy", "Medium", and "Hard". "Mean" denotes the average results of three levels. Note that except for our STNet, the results of other methods are obtained by running the official codes.

	Method	Vehicle				Pedestrian			
	Split	Easy	Medium	Hard	Mean	Easy	Medium	Hard	Mean
	Frame Num.	67832	61252	56647	185731	85280	82253	74219	241752
Success	P2B [50]	57.1	52.0	47.9	52.6	18.1	17.8	17.7	17.9
	BAT [78]	61.0	53.3	48.9	54.7	19.3	17.8	17.2	18.2
	V2B [26]	64.5	55.1	52.0	57.6	27.9	22.5	20.1	23.7
	STNet (ours)	**65.9**	**57.5**	**54.6**	**59.7**	**29.2**	**24.7**	**22.2**	**25.5**
Precision	P2B [50]	65.4	60.7	58.5	61.7	30.8	30.0	29.3	30.1
	BAT [78]	68.3	60.9	57.8	62.7	32.6	29.8	28.3	30.3
	V2B [26]	71.5	63.2	62.0	65.9	43.9	36.2	33.1	37.9
	STNet (ours)	**72.7**	**66.0**	**64.7**	**68.0**	**45.3**	**38.2**	**35.8**	**39.9**

Generalization ability. To verify the generalization ability of our method, we transfer the trained model of the KITTI dataset to obtain the testing results on the nuScenes and Waymo datasets. Following [26], we use the pre-trained models on four categories (car, pedestrian, van, cyclist) of the KITTI dataset to evaluate the corresponding categories (car, pedestrian, truck, bicycle) on the nuScenes dataset. The results are listed in the bottom half of Tab. 1. Note that except for our results, the results of other methods are taken from paper [26]. It can be observed that our method outperforms other methods on the mean results of four categories. In addition, Tab. 2 shows the results of vehicle and pedestrian categories on the Waymo dataset. It can be observed that our method outperforms other methods in terms of different subsets, including easy, medium, and hard. KITTI and Waymo datasets are built by 64-beam LiDAR, while nuScenes dataset is built by 32-beam LiDAR. Due to the large discrepancy between data distributions of datasets caused by different LiDAR sensors and sparsity of point clouds, it is very challenging to directly use the pre-trained model of the KITTI dataset to generalize it on nuScenes. The previous method V2B only achieves the performance of 25.8%/29.0% on the average of four categories. Our method achieves the gains of +1.1%/+1.8% over V2B. However, due to similar data distributions of the KITTI and Waymo datasets, the generalization results of the Waymo dataset are higher than those in the nuScenes dataset. The generalization results further demonstrate the effectiveness of our method for unseen scenes.

Ability to handle sparse scenes. We report the results of different methods in the sparse scenarios. Following [26], we divide the number of points into four intervals, including [0, 150), [150, 1000), [1000, 2500), and [2500, $+\infty$). In Tab. 3, we report average *Success* and *Precision* for each interval on the car category of the KITTI dataset. Note that except for our results, other results are taken from [26]. It can be observed that our method achieves the best performance on all four intervals. Especially in the sparse point clouds below 150

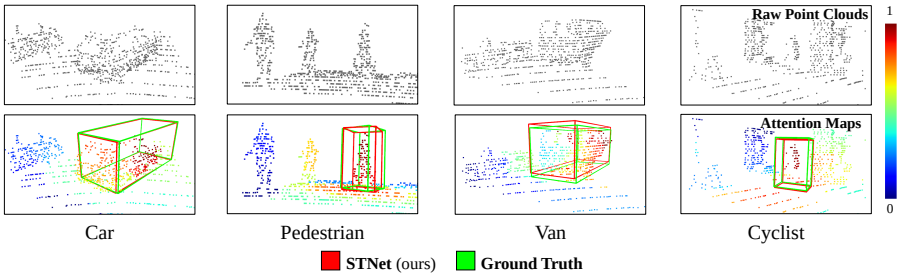

Fig. 3. The attention maps generated by our method on the KITTI dataset.

Table 3. The average *Success* and *Precision* for each interval on the car category in the KITTI dataset.

Method	Success				Precision			
Intervals	[0, 150)	[150, 1k)	[1k, 2.5k)	[2.5k, $+\infty$)	[0, 150)	[150, 1k)	[1k, 2.5k)	[2.5k, $+\infty$)
Frame Num.	3293	2156	693	282	3293	2156	693	282
SC3D [19]	37.9	36.1	33.8	23.7	53.0	53.1	48.7	35.3
P2B [50]	56.0	62.3	51.9	43.8	70.6	78.6	68.1	61.8
BAT [78]	60.7	71.8	69.1	61.6	75.5	83.9	81.0	72.9
V2B [26]	64.7	77.5	72.3	82.2	77.4	87.1	81.5	90.1
STNet (ours)	**66.3**	**77.9**	**79.3**	**83.1**	**79.9**	**87.8**	**89.6**	**91.0**

points, our method can improve the performance by about 2% on both *Success* and *Precision* compared with V2B. Moreover, as the number of points increases, the performance of our method gradually increases. Due to the good results of our method in sparse point clouds, it will lead to more accurate template updates on the subsequent dense frames, resulting in better performance.

Running speed. We also report the average running time of all test frames in the car category on the KITTI dataset. Specifically, we evaluate our model on a single TITAN RTX GPU. Our method achieves 35 FPS, including 4.6 ms for processing point clouds, 22.7 ms for network forward propagation, and 1.3 ms for post-processing. In addition, on the same platform, V2B, P2B and SC3D in default settings run with 37 FPS, 46 FPS, and 2 FPS, respectively. Due to Transformer, the forward time of our method is longer than that in P2B. However, the performance of our method is significantly better than that of P2B.

4.3 Ablation Study

In this section, we conduct the ablation study to validate the effectiveness of the designed modules. Due to a large number of test samples in the car category of the KITTI dataset, the ablated experiments on it can truly reflect the impact of different settings on the tracking accuracy.

Non-local embedding and adaptive interpolation. To verify the effectiveness of our Siamese point Transformer network, which adopts the

non-local feature embedding and the adaptive feature interpolation, we conduct experiments to study their effects on performance. Specifically, we use PointNet++ [49] as the baseline and add the non-local embedding (dubbed "NL emb.") and the adaptive feature interpolation (dubbed "AF inte.") to conduct experiments. The results are listed in Tab. 4. It can be observed that the performance of only PointNet++ is worse than using both non-local feature embedding and adaptive feature interpolation ("NL emb. + AF inte."). According to the results in the ablation study, capturing shape context information of the object can effectively improve tracking performance.

Coarse-to-fine correlation. Here we conduct experiments to study the effects of coarse-to-fine correlation on performance. Specifically, we perform the simple feature augmentation used in P2B [50], only cross-feature augmentation (dubbed "CF aug."), and cross- and ego-feature augmentations (dubbed "CF aug. + EF aug."), respectively. The results are listed in Tab. 4. It can be found that when the cross- and ego-feature augmentations are used at the same time, we can obtain the best results. Since P2B's feature augmentation only uses the cosine distance to measure the similarity between the template and search area, it cannot obtain a high-quality feature fusion map. However, our method iteratively learns the similarity in a coarse-to-fine manner between them, so the target information in the feature fusion map can be further enhanced.

Comparison of attention maps of different components. In Fig. 4, we show the attention maps of the extracted features using our proposed different components. It can be observed that only using the Siamese point Transformer or only using the iterative coarse-to-fine correlation cannot effectively focus on the target object, while STNet using all components is able to effectively distinguish the car from the background. Furthermore, it can be observed that the target car can be clearly recognized from three cars since learned shape information of the target is helpful to learn the discriminative relationship between the template and search area by Transformer.

Table 4. The ablation study results of different components.

Method	*Success*	*Precision*
PointNet++ [49]	66.1	76.9
only NL emb.	69.9	81.8
only AF inte.	68.1	80.9
NL emb. + AF inte.	**72.1**	**84.0**
feature aug. in P2B [50]	69.4	80.8
CF aug.	71.0	82.4
CF aug. + EF aug.	**72.1**	**84.0**

Table 5. The ablation study results of different hyperparameters.

STNet	Parameters	*Success*	*Precision*
Neighbors	$K = 16$	68.1	79.6
	$K = 32$	71.0	82.4
	$K = 48$	**72.1**	**84.0**
	$K = 64$	69.6	81.7
	$K = 80$	68.7	80.5
	$K = 96$	67.2	78.6
Iterations	iter. = 1	69.1	81.7
	iter. = 2	**72.1**	84.0
	iter. = 3	71.8	**84.2**
	iter. = 4	72.0	84.0

Different K in ego-feature augmentation. The neighbor size K is a key parameter in the ego-feature augmentation. Here we study the effects of different values of K on tracking accuracy. In Tab. 5, we report the performance in the cases of different neighbor sizes, including 16, 32, 48, 64, 80, and 96, respectively. It can be observed that when the neighbor size is set to 48, we can obtain the

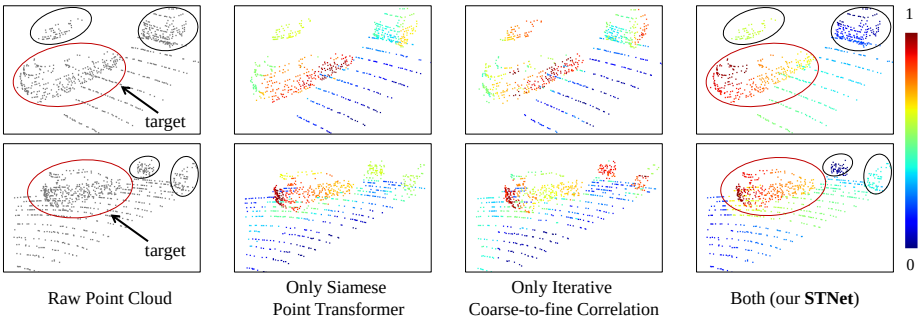

Fig. 4. The attention maps generated by different components of our STNet on the KITTI dataset. The leftmost column shows the input raw point cloud, and we circle the target object to distinguish the background.

best results. If K is too small, our transformer cannot characterize the local geometry structures of point clouds, while if K is too large, background points in the scene are incorporated to localize the target.

Different numbers of iterations. We also study the effects of the number of iterations of coarse-to-fine correlation on performance. In Tab. 5, we list the quantitative results in the cases of different numbers of iterations. It can be observed that the performance of our method with two iterations is comparable to that of our method with three or four iterations. As the number of iterations increases, the GPU memory will gradually increase. Thus, considering the memory consumption, we choose to iterate twice.

5 Conclusions

In this paper, we proposed a 3D Siamese Transformer framework for single object tracking on point clouds. We developed a Siamese point Transformer network that uses the attention mechanism to formulate a encoder-decoder structure to learn shape context information of the target. Also, we constructed an iterative coarse-to-fine correlation network to produce a feature fusion map by using the attention mechanism on the template and the search area. In this way, we can effectively associate the template and the search area in a coarse-to-fine manner so as to mitigate large appearance variations between them in sparse point clouds. The experiments show that the proposed method achieves state-of-the-art performance on the KITTI, nuScenes, and Waymo datasets on 3D single object tracking.

Acknowledgments

The authors would like to thank reviewers for their detailed comments and instructive suggestions. This work was supported by the National Science Fund of China (Grant Nos. U1713208, 61876084).

References

1. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.: Fully-convolutional Siamese networks for object tracking. In: ECCV (2016)
2. Bibi, A., Zhang, T., Ghanem, B.: 3D part-based sparse tracker with automatic synchronization and registration. In: CVPR (2016)
3. Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M.: Visual object tracking using adaptive correlation filters. In: CVPR (2010)
4. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuScenes: A multimodal dataset for autonomous driving. arXiv preprint arXiv:1903.11027 (2019)
5. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with Transformers. In: ECCV (2020)
6. Chen, X., Yan, B., Zhu, J., Wang, D., Yang, X., Lu, H.: Transformer tracking. In: CVPR (2021)
7. Chiu, H.k., Prioletti, A., Li, J., Bohg, J.: Probabilistic 3D multi-object tracking for autonomous driving. arXiv preprint arXiv:2001.05673 (2020)
8. Comport, A.I., Marchand, É., Chaumette, F.: Robust model-based tracking for robot vision. In: IROS (2004)
9. Cui, Y., Fang, Z., Shan, J., Gu, Z., Sifan, Z.: 3D object tracking with Transformer. In: BMVC (2021)
10. Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q.V., Salakhutdinov, R.: Transformer-XL: Attentive language models beyond a fixed-length context. arXiv preprint arXiv:1901.02860 (2019)
11. Danelljan, M., Bhat, G., Shahbaz Khan, F., Felsberg, M.: ECO: Efficient convolution operators for tracking. In: CVPR (2017)
12. Danelljan, M., Hager, G., Shahbaz Khan, F., Felsberg, M.: Learning spatially regularized correlation filters for visual tracking. In: ICCV (2015)
13. Danelljan, M., Shahbaz Khan, F., Felsberg, M., Van de Weijer, J.: Adaptive color attributes for real-time visual tracking. In: CVPR (2014)
14. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
15. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
16. Fan, H., Yang, Y., Kankanhalli, M.: Point 4D Transformer networks for spatio-temporal modeling in point cloud videos. In: CVPR
17. Feng, T., Jiao, L., Zhu, H., Sun, L.: A novel object re-track framework for 3D point clouds. In: ACM MM (2020)
18. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the KITTI vision benchmark suite. In: CVPR (2012)
19. Giancola, S., Zarzar, J., Ghanem, B.: Leveraging shape completion for 3D Siamese tracking. In: CVPR (2019)
20. Gordon, N., Ristic, B., Arulampalam, S.: Beyond the Kalman filter: Particle filters for tracking applications. Artech House, London **830**(5), 1–4 (2004)
21. Guo, M.H., Cai, J.X., Liu, Z.N., Mu, T.J., Martin, R.R., Hu, S.M.: PCT: Point cloud transformer. arXiv preprint arXiv:2012.09688 (2020)

22. Guo, Q., Feng, W., Zhou, C., Huang, R., Wan, L., Wang, S.: Learning dynamic Siamese network for visual object tracking. In: ICCV (2017)
23. Held, D., Thrun, S., Savarese, S.: Learning to track at 100 fps with deep regression networks. In: ECCV (2016)
24. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: Exploiting the circulant structure of tracking-by-detection with kernels. In: ECCV (2012)
25. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE Transactions on Pattern Analysis and Machine Intelligence **37**(3), 583–596 (2014)
26. Hui, L., Wang, L., Cheng, M., Xie, J., Yang, J.: 3D Siamese voxel-to-BEV tracker for sparse point clouds. In: NeurIPS (2021)
27. Hui, L., Yang, H., Cheng, M., Xie, J., Yang, J.: Pyramid point cloud Transformer for large-scale place recognition. In: ICCV (2021)
28. Kart, U., Kamarainen, J.K., Matas, J.: How to make an RGBD tracker? In: ECCV workshops (2018)
29. Kart, U., Lukezic, A., Kristan, M., Kämäräinen, J., Matas, J.: Object tracking by reconstruction with view-specific discriminative correlation filters. In: CVPR (2019)
30. Katharopoulos, A., Vyas, A., Pappas, N., Fleuret, F.: Transformers are RNNs: Fast autoregressive Transformers with linear attention. In: ICML (2020)
31. Kim, A., Ošep, A., Leal-Taixé, L.: EagerMOT: 3D multi-object tracking via sensor fusion. arXiv preprint arXiv:2104.14682 (2021)
32. Kristan, M., Matas, J., Leonardis, A., Felsberg, M., Cehovin, L., Fernandez, G., Vojir, T., Hager, G., Nebehay, G., Pflugfelder, R.: The visual object tracking vot2015 challenge results. In: ICCV workshops (2015)
33. Kristan, M., Matas, J., Leonardis, A., Vojíř, T., Pflugfelder, R., Fernandez, G., Nebehay, G., Porikli, F., Čehovin, L.: A novel performance evaluation methodology for single-target trackers. IEEE Transactions on Pattern Analysis and Machine Intelligence **38**(11), 2137–2155 (2016)
34. Lee, K.H., Hwang, J.N.: On-road pedestrian tracking across multiple driving recorders. IEEE Transactions on Multimedia **17**(9), 1429–1438 (2015)
35. Lin, Z., Feng, M., Santos, C.N.d., Yu, M., Xiang, B., Zhou, B., Bengio, Y.: A structured self-attentive sentence embedding. arXiv preprint arXiv:1703.03130 (2017)
36. Liu, Y., Jing, X.Y., Nie, J., Gao, H., Liu, J., Jiang, G.P.: Context-aware three-dimensional mean-shift with occlusion handling for robust object tracking in RGB-D videos. IEEE Transactions on Multimedia **21**(3), 664–677 (2018)
37. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin Transformer: Hierarchical vision Transformer using shifted windows. In: ICCV (2021)
38. Liu, Z., Chen, W., Lu, J., Wang, H., Wang, J.: Formation control of mobile robots using distributed controller with sampled-data and communication delays. IEEE Transactions on Control Systems Technology **24**(6), 2125–2132 (2016)
39. Liu, Z., Suo, C., Liu, Y., Shen, Y., Qiao, Z., Wei, H., Zhou, S., Li, H., Liang, X., Wang, H., et al.: Deep learning-based localization and perception systems: approaches for autonomous cargo transportation vehicles in large-scale, semiclosed environments. IEEE Robotics and Automation Magazine **27**(2), 139–150 (2020)
40. Luber, M., Spinello, L., Arras, K.O.: People tracking in RGB-D data with on-line boosted target models. In: IROS (2011)

41. Luo, W., Yang, B., Urtasun, R.: Fast and furious: Real time end-to-end 3D detection, tracking and motion forecasting with a single convolutional net. In: CVPR (2018)
42. Lüscher, C., Beck, E., Irie, K., Kitza, M., Michel, W., Zeyer, A., Schlüter, R., Ney, H.: RWTH ASR Systems for LibriSpeech: Hybrid vs attention–w/o data augmentation. arXiv preprint arXiv:1905.03072 (2019)
43. Mao, J., Xue, Y., Niu, M., Bai, H., Feng, J., Liang, X., Xu, H., Xu, C.: Voxel Transformer for 3D object detection. In: ICCV (2021)
44. Pan, X., Xia, Z., Song, S., Li, L.E., Huang, G.: 3D object detection with pointformer. In: CVPR (2021)
45. Pang, Z., Li, Z., Wang, N.: Model-free vehicle tracking and state estimation in point cloud sequences. In: IROS (2021)
46. Pieropan, A., Bergström, N., Ishikawa, M., Kjellström, H.: Robust 3D tracking of unknown objects. In: 2015 IEEE International Conference on Robotics and Automation (ICRA). pp. 2410–2417. IEEE (2015)
47. Qi, C.R., Litany, O., He, K., Guibas, L.J.: Deep hough voting for 3D object detection in point clouds. In: ICCV (2019)
48. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: Deep learning on point sets for 3D classification and segmentation. In: CVPR (2017)
49. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: PointNet++: Deep hierarchical feature learning on point sets in a metric space. In: NeurIPS (2017)
50. Qi, H., Feng, C., Cao, Z., Zhao, F., Xiao, Y.: P2B: Point-to-box network for 3D object tracking in point clouds. In: CVPR (2020)
51. Scheidegger, S., Benjaminsson, J., Rosenberg, E., Krishnan, A., Granström, K.: Mono-camera 3D multi-object tracking using deep learning detections and PMBM filtering. In: IV (2018)
52. Shan, J., Zhou, S., Fang, Z., Cui, Y.: PTT: Point-track-Transformer module for 3D single object tracking in point clouds. In: IROS (2021)
53. Shenoi, A., Patel, M., Gwak, J., Goebel, P., Sadeghian, A., Rezatofighi, H., Martín-Martín, R., Savarese, S.: JRMOT: A real-time 3D multi-object tracker and a new large-scale dataset. In: IROS (2020)
54. Shi, S., Wang, X., Li, H.: PointRCNN: 3D object proposal generation and detection from point cloud. In: CVPR (2019)
55. Shi, S., Wang, Z., Shi, J., Wang, X., Li, H.: From points to parts: 3D object detection from point cloud with part-aware and part-aggregation network. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)
56. Spinello, L., Arras, K., Triebel, R., Siegwart, R.: A layered approach to people detection in 3D range data. In: AAAI (2010)
57. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: CVPR (2020)
58. Synnaeve, G., Xu, Q., Kahn, J., Likhomanenko, T., Grave, E., Pratap, V., Sriram, A., Liptchinsky, V., Collobert, R.: End-to-end ASR: from supervised to semi-supervised learning with modern architectures. arXiv preprint arXiv:1911.08460 (2019)
59. Tang, S., Andriluka, M., Andres, B., Schiele, B.: Multiple people tracking by lifted multicut and person re-identification. In: CVPR (2017)
60. Tao, R., Gavves, E., Smeulders, A.W.: Siamese instance search for tracking. In: CVPR (2016)
61. Valmadre, J., Bertinetto, L., Henriques, J., Vedaldi, A., Torr, P.H.: End-to-end representation learning for correlation filter based tracking. In: CVPR (2017)

62. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. arXiv preprint arXiv:1706.03762 (2017)
63. Wang, Q., Gao, J., Xing, J., Zhang, M., Hu, W.: DCFNet: Discriminant correlation filters network for visual tracking. arXiv preprint arXiv:1704.04057 (2017)
64. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision Transformer: A versatile backbone for dense prediction without convolutions. In: ICCV (2021)
65. Wang, Y., Weng, X., Kitani, K.: Joint detection and multi-object tracking with graph neural networks. arXiv preprint arXiv:2006.13164 (2020)
66. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. arXiv preprint arXiv:1801.07829 (2018)
67. Wang, Z., Xie, Q., Lai, Y.K., Wu, J., Long, K., Wang, J.: MLVSNet: Multi-level voting Siamese network for 3D visual tracking. In: ICCV (2021)
68. Weng, X., Wang, J., Held, D., Kitani, K.: 3D multi-object tracking: A baseline and new evaluation metrics. In: IROS (2020)
69. Weng, X., Wang, Y., Man, Y., Kitani, K.M.: GNN3DMOT: Graph neural network for 3D multi-object tracking with 2D-3D multi-feature learning. In: CVPR (2020)
70. Weng, X., Yuan, Y., Kitani, K.: Joint 3D tracking and forecasting with graph neural network and diversity sampling. arXiv preprint arXiv:2003.07847 (2020)
71. Wu, H., Han, W., Wen, C., Li, X., Wang, C.: 3D multi-object tracking in point clouds based on prediction confidence-guided data association. IEEE Transactions on Intelligent Transportation Systems (2021)
72. Xing, J., Ai, H., Lao, S.: Multiple human tracking based on multi-view upper-body detection and discriminative learning. In: ICPR (2010)
73. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., Le, Q.V.: XLNet: Generalized autoregressive pretraining for language understanding. arXiv preprint arXiv:1906.08237 (2019)
74. Yin, T., Zhou, X., Krahenbuhl, P.: Center-based 3D object detection and tracking. In: CVPR (2021)
75. Zhang, M., Xing, J., Gao, J., Shi, X., Wang, Q., Hu, W.: Joint scale-spatial correlation tracking with adaptive rotation estimation. In: ICCV workshops (2015)
76. Zhang, W., Zhou, H., Sun, S., Wang, Z., Shi, J., Loy, C.C.: Robust multi-modality multi-object tracking. In: CVPR (2019)
77. Zhao, H., Jiang, L., Jia, J., Torr, P., Koltun, V.: Point Transformer. In: ICCV (2021)
78. Zheng, C., Yan, X., Gao, J., Zhao, W., Zhang, W., Li, Z., Cui, S.: Box-aware feature enhancement for single object tracking on point clouds. In: ICCV (2021)
79. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable DETR: Deformable Transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)
80. Zhu, Z., Wang, Q., Li, B., Wu, W., Yan, J., Hu, W.: Distractor-aware Siamese networks for visual object tracking. In: ECCV (2018)