Space-Correlated Transformer: Jointly Explore the Matching and Motion Clues in 3D Single Object Tracking

Fei Xie[1⋆], Jiahao Nie[2⋆], Zhongdao Wang[3], Zhiwei He[2†], and Chao Ma[1†]

[1] MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
[2] Hangzhou Dianzi University
[3] Huawei Noah's Ark Lab

Abstract. 3D Single Object Tracking (3D SOT) in LiDAR point clouds play a crucial role in autonomous driving. Current approaches mostly follow two paradigms, i.e., Siamese matching-based and motion-centric. However, LiDAR point clouds lack enough appearance information, while the motion-centric trackers suffer from complex model structures. To address these issues, we present a novel and conceptually simple tracking framework dubbed SCtrack, which jointly explores the matching and motion clues in point clouds. Specifically, SCtrack embeds point clouds into spatially structured features and conducts space correlation along the aligned spatial region. The target relative motion is directly inferred from the correlated features. In contrast to prevalent PointNet-based features, our spatially structured representation inherently models motion clues among the consecutive frames of point clouds, thereby being complementary to appearance matching. To better utilize the aligned structured features, we employ a strategy of varied-size space regions that adapt to different target shapes and locations during space correlation. Without bells and whistles, SCtrack achieves leading performance, with 89.1%, 71.5%, and 62.7% precision on KITTI, NuScenes, and Waymo Open Dataset, and runs at a considerably high speed of 60 Fps on a single RTX3090 GPU. Extensive studies validate the effectiveness of our SCtrack framework. The code and models will be released.

Keywords: Structured representation · Space-correlation · Point Cloud · 3D Single Object Tracking

1 Introduction

3D single object tracking (SOT) based on point clouds is a fundamental task with enormous potential for various applications, including autonomous driving and robotics [38, 43, 77, 101]. Early efforts [2, 15, 36, 37, 40, 79, 91] 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

⋆ Equal contribution † Corresponding anthor

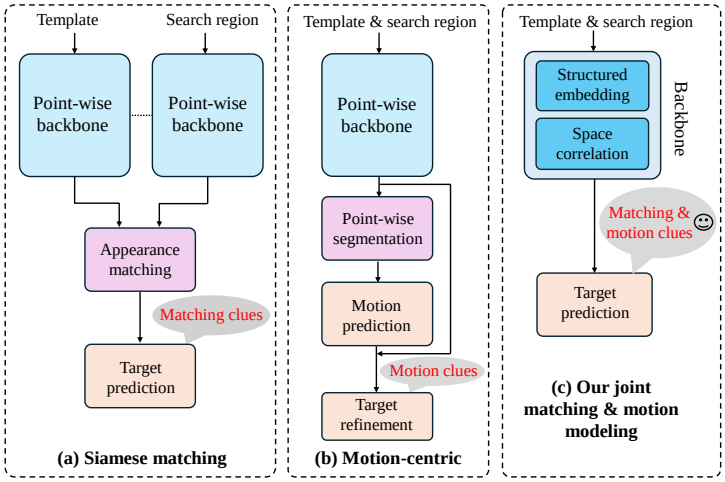

Fig. 1: Comparison with typical 3D SOT paradigms. (a) Siamese matching paradigm [23,31,64,84] adopts point-wise representation [63] and conducts appearance-matching operations, neglecting point clouds' insufficient appearance and shape information. (b) motion-centric [96] leverage motion modeling while suffering complex two-stage model structure. (c) On the contrary, our proposed SCtrack paradigm jointly explores appearance and matching information, greatly retaining the spatial information and remaining a conceptually simple tracking pipeline.

for 3D object tracking. 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. As a result, accurately and efficiently analyzing point cloud data to track objects in complex scenes is still a challenging and open problem.

Many current tracking approaches [12, 30, 31, 67, 76, 95] use a point-wise representation by taking raw point clouds as input. For instance, P2B [64] and its subsequent works use a point-based network with a Siamese architecture for feature extraction. This is followed by a point-wise appearance-matching module that propagates target cues and a target prediction network, as shown in Fig. 1 (a). M2-Track [96] proposes a motion-centric approach that first identifies the target points by segmenting them from their surroundings using a PointNet [62] segmentation network. Then, it leverages motion clues to refine the localization of the target, as shown in Fig. 1 (b). Although these approaches [12, 21, 23, 30, 31, 64, 67, 76, 95] have shown excellent performance in tracking benchmarks, motion-centric trackers often necessitate complex model structures. In the meantime, Siamese matching trackers suffer from insufficient appearance information in point-wise features. Therefore, A natural question arises: can we simultaneously leverage both motion and appearance clues without increasing model complexity?

In this paper, we present a novel and conceptually simple tracking paradigm, dubbed SCtrack, which jointly explores appearance matching and motion clues for 3D SOT, as shown in Fig. 1 (c). To ensure efficiency and simplicity, we com-

bine the separated feature extraction and matching in the conventional Siamese paradigm and remove the two stages of the motion modeling paradigm with one single network. Our proposed backbone network consists of two core parts: the structured embedding module and the Space-Correlated Transformer (SCT) module. In contrast to the widely adopted point-wise feature networks, *e.g.*, PointNet [62] and PointNet++ [63], our structured embedding module adopts spatially aligned structure representations for better appearance matching and motion modeling. Specifically, to exploit the spatial relations among tracked targets and distractors, we subdivide the 3D space into equally spaced voxels given a pair of point clouds from the template and search regions. With structured voxel features, we squeeze, concatenate, and merge the template and search features along the channel dimension. Then, a space-correlated transformer is proposed to exploit the motion and appearance clues on the structured features. SCT partitions the merged template and search features into different space regions for correlation. Intuitively, using a fixed-size region may be sub-optimal for dealing with targets of different sizes. To better adapt to the diverse shapes and motion patterns of the tracked targets, we use a tiny hyper-network module to predict the size and location of the correlated space regions. The hyper-network module can learn the diverse correlated region size depending on the template and search region features as inputs. Thus, the rich motion and appearance clues among two consecutive frames can be well-captured, while the feature correlation can also fulfill the appearance matching. In the final, we can leverage a single backbone network, which includes the structured embedding module and space-correlated transformer, to realize synchronized matching and motion modeling.

In conclusion, the contribution of this study is threefold as follows:

- We present a joint matching and motion modeling tracking paradigm, which overcomes the inability of the traditional Siamese paradigm and is conceptually simpler than the motion-centric paradigm in the model pipeline.
- We adopt spatially structured representation and propose a space-correlated transformer with adaptive region size, greatly advancing the feature learning of 3D single object tracking.
- Our joint matching and motion modeling method, SCtrack, achieves state-of-the-art performance on the KITTI, nuScenes, and Waymo datasets in 3D single object tracking.

2 Related Work

2D Single Object Tracking. In the context of 2D single object tracking, Siamese-based [9, 39, 40, 51, 54, 82, 88, 94] and transformer-based [7, 13, 57, 78–81, 89, 91] trackers, which have attained great attention for their dominant performance and speed. Siamese trackers, including SiamFC [1], SiamRPN [40], SiamCAR [24], STARK [89], SBT [79] and SuperSBT [83] formulate the tracking problem as an appearance-matching problem between the template and search images. Despite the great success of Siamese techniques in 2D single object

tracking, such techniques encounter difficulty in 3D single object tracking as the point clouds lack appearance information, e.g. texture, shape, and background contexts. It is worth noting that few 2D single object tracking methods consider exploiting motion information due to the complex motion patterns in natural 2D image data.

3D Single Object Tracking. 3D Single Object Tracking mostly draw the inspirations from 2D tracking approaches [4, 28]. Recently, Siamese trackers [26, 27, 71, 73, 100] have significantly improved tracking performance compared to the traditional correlation filtering based trackers [16, 17, 29, 92]. Previous methods [3, 46, 60, 68] adopt RGB-D data for 3D single object tracking. RGB-D based trackers [34, 35, 44] heavily rely on RGB data and adopt similar model pipelines used in 2D visual tracking. Recently, researchers [23, 52, 53, 56, 59] have focused on using 3D point clouds for single object tracking. P2B [64] developes a 3D region proposal network that builds on the success of SiamRPN [40] and VoteNet [61]. It has significantly improved tracking performance and become a strong baseline. Then, SC3D [23] proposes a shape completion-based 3D Siamese tracker for 3D tracking. It performs template matching between the template and plenty of candidate proposals in the search area. Lately, to handle sparse point clouds, V2B [30] proposed a Siamese voxel-to-BEV tracker, which contains a Siamese shape-ware feature learning network and a voxel-to-BEV target localization network. In recent years, the Transforemer are introduced into follow-up methods [31, 67, 95, 97]. For example, some methods [55, 86] have incorporated a global-local transformer module to improve proposal quality, while others have designed a target-centric transformer network to explore contextual information. MBPTrack [87] further improves these methods by introducing a memory network and a box-prior localization network. Moreover, SyncTrack [48] simplifies the “Extracting then Matching" pipeline for Siamese tracking by synchronizing feature extraction and matching in a one-stream framework. Unlike the Siamese matching-based paradigm, M2Track [96] introduces a motion-centric paradigm to exploit motion clues. It models the relationship between consecutive frames using two model stages: target segmentation and target prediction refinement.

Despite their great success, current 3D Siamese matching trackers develop complex matching modules due to the lack of motion modeling ability. It mainly attributes to that Point clouds are incomplete and lack texture. Additionally, 3D tracking only predicts pose parameters instead of shape, which makes motion modeling a critical factor. In this work, we joint exploit appearance matching and motion modeling by constructing a conceptually simple framework, which fully leverages spatially structured feature representation to guide accurate target localization.

Transformer and attention. Transformer is first introduced in [72], which uses a self-attention mechanism [42] 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 processing [14, 18, 90], speech processing [47, 70]. Recently, ViT [19] first proposed a vision Transformer for image recognition, introducing a transformer to handle

visual tasks. After that, the transformer is extended to various visual tasks, such as semantic segmentation [45, 74], object detection [6, 93, 99], object tracking [8]. Lately, inspired by point transformer, different 3D vision tasks apply transformer to yield good performance, such as point cloud classification [25], point cloud-based place recognition [32], 3D object detection [49, 58], 3D object tracking [12, 67], and 3D action recognition [20].

In 3D SOT, several transformer tracking methods [31, 55, 67, 97] have been proposed. They either use self-attention to process features or use cross-attention to interact with features from the template and search regions for appearance matching. In contrast, we leverage the transformer scheme within the structured space region to exploit motion and appearance matching jointly.

3 Method

3.1 Problem Definition

In the configuration of the 3D LiDAR single object detection (SOT) task, the 3D bounding box (BBox) is defined as $(x, y, z, w, l, h, \theta) \in \mathbb{R}^7$, where the (x, y, z) represents the coordinate center of the BBox and $(w, l, h), \theta$ stand for the BBox size and heading angle (the rotation around the *up-axis*) respectively. Generally, the BBox size is assumed to be fixed by default even when the target object is non-rigid, thus minimizing the dimensions of BBox from $\mathbb{R}^7$ to $\mathbb{R}^4$. Given a sequence of temporally-connected point clouds $\{\mathcal{P}_i\}_{i=1}^{T}$ (T is the number of points in each frame) and an initial BBox $\mathcal{B}_1$ of the target, the goal of SOT is to localize the target BBoxes $\{\mathcal{B}_i\}_{i=2}^{T}$ in all frames online. Following the previous manner, a template point cloud $\mathcal{P}^t = \{p_i^t\}_{i=1}^{N_t}$ and a search region $\mathcal{P}^s = \{p_i^s\}_{i=1}^{N_s}$ are generated, where N_t and N_s are number of template and search region points. The template P^t is generated by cropping and centering the target in the initial frame based on the initial BBox.

3.2 Joint Matching and Motion Modeling Pipeline

We propose to unify the independent feature extraction and matching in the Siamese paradigm and remove the two stages of the motion-centric paradigm with one single backbone. Within a backbone network, we first use the structured embedding module to process unstructured points. Then, we adopt a space-correlated transformer for feature correlation and motion modeling. Finally, the target position is directly predicted from the extracted features. Our SCtrack can realize synchronized matching and motion modeling.

Spatially structured embedding. The proposed Spatially structured network consists of two functional processes: (1) Voxel generation and (2) Voxel feature encoding, as illustrated in Fig. 2 and Fig.3. We provide a detailed introduction to the spatially structured module as follows:

To accurately capture the relative motion of the tracked target, we subdivide the 3D space into equally spaced voxels given a pair of point clouds from the

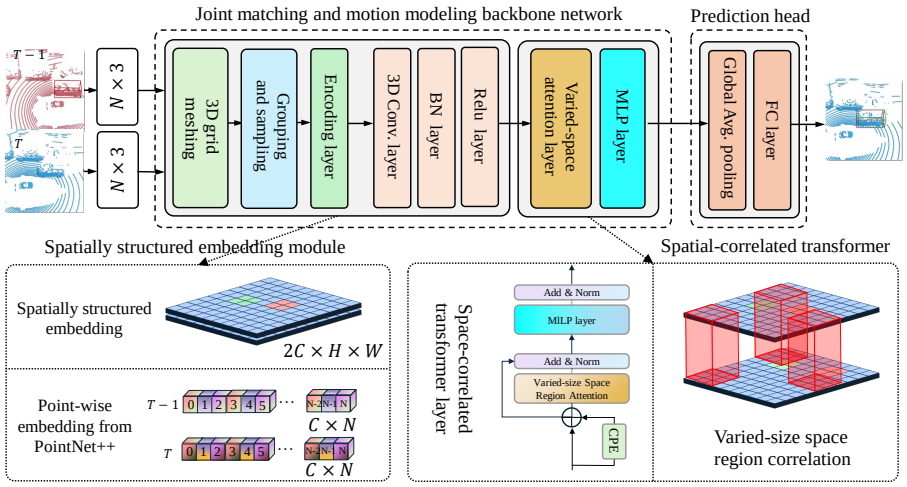

Fig. 2: The detailed architecture of our joint matching and motion modeling tracking pipeline. Our simple tracking paradigm comprises a backbone network for joint modeling and a prediction head for directly outputting the relative target motion. The proposed feature network consists of a spatially structured embedding module and a space-correlated transformer. FC layer denotes the fully connected layers.

template and search regions. Suppose the point cloud encompasses 3D space with range D, H, W along the z, y, x axes respectively. We define each voxel of size v_D, v_H, and v_W accordingly. The resulting 3D voxel grid is of size $D' = D/v_D, H' = H/v_H, W' = W/v_W$. Due to the sparsity and highly variable point density throughout the space, we group the points according to the voxel they are located in. Then, we randomly sample a fixed number, N, of points from those voxels containing more than N points. To encode the shape of the surface contained within the voxel, we adopt the hierarchical feature encoding layer in VoxelNet [98] to process each voxel. Then, we obtain voxel features $\mathbf{VF}_{\{C,D,H,W\}}$ with the grid size $\{v_D, v_H, v_W\}$ and channel dimension C, repectively. We further feed voxel features $\mathbf{VF}_{\{C,D,H,W\}}$ into 3D convolutional layers, which consist of stacked 3D convolution, BN [33] layer, and ReLU [66] layer. The 3D convolutional layers aggregate voxel-wise features within a progressively expanding receptive field, adding more context to the shape description. The detailed architectural variants of the embedding module are in supplementary materials.

Comparisons with point-wise feature embedding. The mainstream 3D single object trackers commonly adopt point-wise feature representation extracted by PointNet [62, 63]. However, the PointNet-based features emphasize appearance matching more while neglecting the spatial relationship among points. In contrast, our spatially structured representation inherently reflects the spatial relationships among corresponding points, thereby guiding a more accurate relative motion of the target between two consecutive frames for 3D tracking tasks.

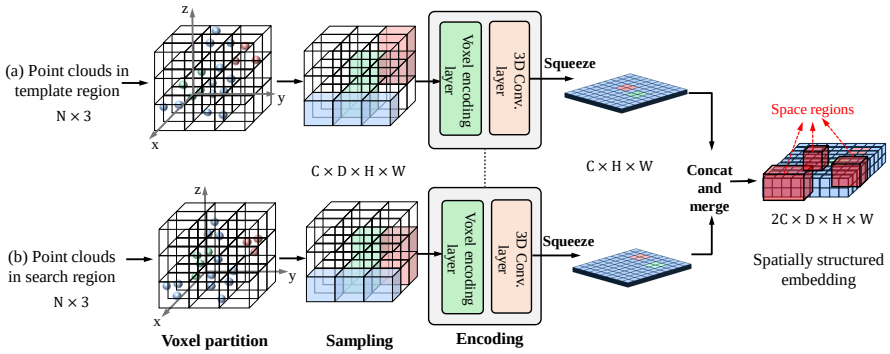

Fig. 3: The details of spatially structured embedding module. The structured features of the template and search region are squeezed, concatenated, and merged to capture the target motion clues. In contrast to the popular point-wise features, spatially structured feature representation can preserve more spatial information for predicting relative target motion in two consecutive frames.

3.3 Space-Correlated Transformer

Space-correlated transformer (SCT) is built on top of the basic structure of the transformer layer and serves as a feature correlation module for the template and search region, as shown in Fig. 3. After obtaining the embedding $\mathbf{VF}_{\{C,D,H,W\}}$, we use SCT to capture the appearance and motion clues within a structured space region (see red cubes in Fig.3). Rather than using fixed-size space region to conduct spatially structured correlation, our SCT learns to flexibly determine the space region size, *i.e.*, spatially aligned space area, given template/search region structured input.

Varied-size space generation. Technically, given the voxel-based structured features $\mathbf{VF}^{\text{temp}}_{\{C,D,H,W\}}$ and $\mathbf{VF}^{\text{search}}_{\{C,D,H,W\}}$, SCT first squeezes structured features along the height dimension and concatenates the template and search region features along the channel dimension. Then, SCT merges structured features along the spatial dimension, *i.e.*, the XOY plane. Thus, we spatially align the structured features of the template and search region, which are for the next-step joint modeling:

$$\begin{aligned}
\mathbf{VF}^{\text{temp}}_{\{C,H,W\}}, \mathbf{VF}^{\text{search}}_{\{C,H,W\}} &= \mathbf{Squeeze}(\mathbf{VF}^{\text{temp}}_{\{C,D,H,W\}}, \mathbf{VF}^{\text{search}}_{\{C,D,H,W\}}), \\
\mathbf{VF}^{\text{all}}_{\{C+C,H,W\}} &= \mathbf{Concat}(\mathbf{VF}^{\text{temp}}_{\{C,H,W\}}, \mathbf{VF}^{\text{search}}_{\{C,H,W\}}), \\
\mathbf{VF}^{\text{all}}_{\{2C,H,W\}} &= \mathbf{Merge}(\mathbf{VF}^{\text{all}}_{\{C+C,H,W\}}),
\end{aligned} \tag{1}$$

where $\{\mathbf{Squeeze}, \mathbf{Concat}, \mathbf{Merge}\}$ denotes the squeeze, concatenation and merge operation. Since each feature vector in $\mathbf{VF}^{\text{all}}_{\{2C,H,W\}}$ spatially aligns the features of the template and search region, we partition the feature into space regions $\text{VF}_{R|\{1,2,...,M\}}$ with the M predefined regions in XOY plane. We refer to these regions as default regions, and details are in supplementary materials.

To better adapt to the diverse target shapes and motion patterns, it is critical to transform default regions into varied-size regions. As shown in Fig. 4, we adopt

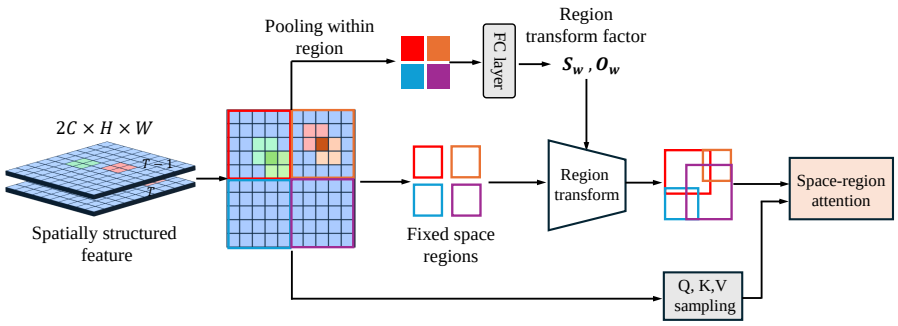

Fig. 4: The details of varied-size space region correlation transformer module, SCT. Varied-size space regions are learned to capture the motion clues of the structured features and adapted to different target sizes.

two transforming factors, *i.e.*, scale factor S_r to modulate the spatial size, and offset factor O_r to translate the default regions in the horizontal and vertical direction. Inspired by the [50, 65, 93], we use a tiny hyper-network to predict these two transforming factors:

$$S_{r|\{1,2,...,M\}}, O_{r|\{1,2,...,M\}} = \textbf{HyperNetwork}(\textbf{VF}^{\text{default}}_{R|\{1,2,...,M\}}), \tag{2}$$

the hyper-network is highly efficient, which only consists of an average pooling layer, a LeakyReLU [85] activation layer, and a 1×1 fully-connected layer with stride 1. Then, we obtain feature partitions in varied-size regions:

$$\textbf{VF}^{\text{varied}}_{R|\{1,2,...,M\}} = \textbf{Transform}(\textbf{VF}^{\text{default}}_{R|\{1,2,...,M\}}|S_{r|\{1,2,...,M\}}, O_{r|\{1,2,...,M\}}), \tag{3}$$

then, the feature correlation is conducted within the generated regions.

Space-correlated attention. Once varied-size space regions are generated, we conduct correlation to capture the motion and appearance clues between template and search regions. We first get the query features from the default regions using a linear projection layer, *i.e.*,

$$Q_r = \textbf{LinearProject}(\textbf{VF}^{\text{default}}_{R|\{1,2,...,M\}}). \tag{4}$$

As the hyper-network has a restricted receptive field, we use the query features from the default regions and key/value features from the corresponding varied-size regions to build cross-region dependencies. Then, we sample the key and value tokens $\{K_r, V_r\}$ from the learned varied-size regions, *i.e.*,

$$K_r, V_r = \textbf{LinearProject}(\textbf{VF}^{\text{default}}_{R|\{1,2,...,M\}}|S_{r|\{1,2,...,M\}}, O_{r|\{1,2,...,M\}}). \tag{5}$$

The sampled vectors K_r, V_r are then fed into Multi-Head Self Attention (MHSA) [19] with queries Q_r for attention computation:

$$Q^{\text{attn}}_r = \textbf{MHSA}(Q_r|(K_r, V_r). \tag{6}$$

However, as the key/value vectors are sampled from different locations with the query vectors, the relative position embeddings between the query and key vectors may not describe the spatial relationship well. Following [10, 75], we further

adopt conditional position embedding [11] (CPE) before the MHSA layers, *i.e.*,

$$\mathbf{VF} = \mathbf{VF}^{l-1} + \mathbf{CPE}(\mathbf{VF}^{l-1}), \tag{7}$$

where $\mathbf{VF}^{l-1}$ is the structured feature from the previous transformer layer. CPE is implemented by a depth-wise convolution layer. Details can be found in supplementary materials.

Comparisons with 3D Siamese tracking. Siamese-based trackers mostly follow the "Extracting then Matching" paradigm and neglect the motion clues in 3D tracking. However, the insufficient appearance information in point clouds hinders Siamese trackers from effective object tracking. Motion-centric trackers, *e.g.*, M2Track [96] suffer from complex model structures and multi-stage training. In contrast, our joint matching and motion framework achieves a win-win scenario: it allows for effective motion modeling and adopts a pseudo-Siamese structure, which is much simpler than the motion-centric model pipeline.

3.4 Prediction Head and Loss

Prediction head. In contrast to the complex prediction head in motion-centric tracker M2Track that introduces an additional motion classifier, we follow the simple head design in the Siamese tracker. We apply a max-pooling operation directly to the correlated feature $\mathbf{VF}^{\text{all}}_{\{2C,H\times W\}} \in \mathcal{R}^{D\times C}$ and directly infer 4 Degree-Of-Freedom relative motion of the target:

$$\mathcal{M}_{t-1,t} = \mathbf{MLP}\left(\mathbf{Maxpooling}\left(\mathbf{VF}^{\text{all}}_{\{2C,H\times W\}}\right)\right), \tag{8}$$

where $\mathcal{M}_{t-1,t} = (\Delta x_t, \Delta y_t, \Delta z_t, \Delta\theta_t)$ is to transform the predicted box.

Training phase. Our SCtrack model can be trained end-to-end by directly predicting the relative 4-DOF motion state. To capture diverse motion distributions, we adopt residual log-likelihood estimation regression loss [41] for the training phase. More details can be found in supplementary materials.

Inference phase. During the inference phase, the tracker predicts a series of relative target motion states $\{\mathcal{M}_{t-1,t}\}_{t=1}^{T}$ for each frame in a point cloud sequence. With the predicted target motion, we apply a rigid body transformation [31, 86, 96], which is commonly adopted by most 3D trackers, to obtain the target location in the current timestamp:

$$\mathcal{B}_t = \mathbf{F}_{\text{rbt}}\ (\mathcal{B}_{t-1}, \mathcal{M}_{t-1,t}), \tag{9}$$

where a rigid body transformation $\mathbf{F}_{\text{rbt}}$ is to conduct to obtain the current target position $\mathcal{B}_t$ from the previous result $\mathcal{B}_{t-1}$.

4 Experiments

4.1 Experimental Settings

Implementation details. To train the SCtrack model, we use the AdamW optimizer on two Tesla A800 GPUs. The batch size is set to 32. We set the

Table 1: The performance of different methods on the KITTI datasets. "Mean" denotes the average results of four categories.

	Method	Success					Precision				
	Category	Car	Pedestrian	Van	Cyclist	Mean	Car	Pedestrian	Van	Cyclist	Mean
	Frame Num.	6424	6088	1248	308	14068	6424	6088	1248	308	14068
KITTI	SC3D [23]	41.3	18.2	40.4	41.5	31.2	57.9	37.8	47.0	70.4	48.5
	P2B [64]	56.2	28.7	40.8	32.1	42.4	72.8	49.6	48.4	44.7	60.0
	MLVSNet [76]	56.0	34.1	52.0	34.3	45.7	74.0	61.1	61.4	44.5	66.6
	LTTR [12]	65.0	33.2	35.8	66.2	48.7	77.1	56.8	45.6	89.9	65.8
	BAT [95]	60.5	42.1	52.4	33.7	51.2	77.7	70.1	67.0	45.4	72.8
	PTTR [97]	65.2	50.9	52.5	65.1	58.4	77.4	81.6	61.8	90.5	77.8
	PTT [67]	67.8	44.9	43.6	37.2	55.1	81.8	72.0	52.5	47.3	74.2
	V2B [30]	70.5	48.3	50.1	40.8	58.4	81.3	73.5	58.0	49.7	75.2
	STNet [31]	72.1	49.9	58.0	73.5	61.3	84.0	77.2	70.6	93.7	80.1
	BAT [95]	60.5	42.1	52.4	33.7	51.2	77.7	70.1	67.0	45.4	72.8
	GLT-T [55]	68.2	52.4	52.6	68.9	60.1	82.1	78.8	62.9	92.1	79.3
	CXTrack [86]	69.1	67.0	60.0	74.2	67.5	81.6	91.5	71.8	94.3	85.3
	SyncTrack [48]	73.3	54.7	60.3	73.1	64.1	85.0	80.5	70.0	93.8	81.9
	MBPTrack [87]	73.4	68.6	61.3	76.7	70.3	84.8	93.9	72.7	94.3	87.9
	SCtrack	**73.8**	**68.6**	**70.7**	**75.2**	**71.5**	**85.4**	**92.9**	**83.8**	**94.4**	**89.1**

initial learning rate to 10^{-4} and decay it by a factor of 5 every 40 epochs. To construct the point cloud inputs for two consecutive frames at timestamps $t-1$ and t, we crop search regions that are located around the previous target within a range of $[(x_{\min}, x_{\max}), (y_{\min}, y_{\max}), (z_{\min}, z_{\max})]$. For cars and humans categories, the ranges are set as $[\pm 4.8, \pm 1.5, \pm 1.92, \pm 1.5]$ to contain relevant points, respectively. To improve the model's robustness, we include simulated test error strategy for consecutive frames $t-1$ and t during training. This strategy involves the random horizontal flipping of points and bounding boxes of the targets, along with uniform rotations around their vertical axis within $[-5°, 5°]$. We add random translations along the x, y, and z axes to targets in the current frame t to simulate motion patterns and enhance model accuracy. The translations are sampled as Gaussian distribution with parameters $[\mu, \sigma]$. Considering the data prior, the target moves mostly along the x axis. We set the Mean μ and variance σ for x, y, and z axes to $[0, 0.3]$, $[0, 0.1]$, and $[0, 0.1]$, respectively, which simulate the target movement better.

Datasets. We adopt three widely-used 3D SOT datasets to evaluate our methods, *i.e.*, KITTI [22], nuScenes [5], and Waymo [69]. The KITTI dataset contains 21 video sequences, which we split into three parts as follows: sequences $0-16$ for training, $17-18$ for validation, and $19-20$ for testing, based on [23]. The nuScenes dataset comprises 700 sequences for training and 150 sequences for validation. However, the ground truth for the test set in nuScenes is unavailable offline, so we use its validation set to evaluate our method. For the Waymo dataset, we adopt $1,121$ tracklets based on LiDAR-SOT [59]. The tracklets have been split into easy, medium, and hard subsets, depending on the number of points in the first frame of each tracklet. Following [30], we use the model trained on the KITTI dataset to test the nuScenes and Waymo datasets to evaluate the generalization ability of our 3D tracker.

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

	Method	Vehicle				Pedestrian				
	Split	Easy	Medium	Hard	Mean	Easy	Medium	Hard	Mean	Mean
	Frame Num.	67832	61252	56647	185731	85280	82253	74219	241752	427483
Success	P2B [64]	57.1	52.0	47.9	52.6	18.1	17.8	17.7	17.9	33.0
	BAT [95]	61.0	53.3	48.9	54.7	19.3	17.8	17.2	18.2	34.1
	V2B [30]	64.5	55.1	52.0	57.6	27.9	22.5	20.1	23.7	38.4
	STNet [31]	65.9	57.5	54.6	59.7	29.2	24.7	22.2	25.5	40.4
	CXTrack [86]	63.9	54.2	52.1	57.1	35.4	29.7	26.3	30.7	42.2
	MBPTrack [87]	**68.5**	**58.4**	**57.6**	**61.9**	37.5	33.0	30.0	33.7	46.0
	SCtrack	66.1	57.2	56.6	59.9	**43.4**	**36.6**	**31.1**	**46.9**	**46.8**
Precision	P2B [64]	65.4	60.7	58.5	61.7	30.8	30.0	29.3	30.1	43.8
	BAT [95]	68.3	60.9	57.8	62.7	32.6	29.8	28.3	30.3	44.4
	V2B [30]	71.5	63.2	62.0	65.9	43.9	36.2	33.1	37.9	50.1
	STNet [31]	72.7	66.0	64.7	68.0	45.3	38.2	35.8	39.9	52.1
	CXTrack [86] [86]	71.1	62.7	63.7	66.1	55.3	47.9	44.4	49.4	56.7
	MBPTrack [87] [87]	**77.1**	**68.1**	**69.7**	**71.9**	57.0	51.9	48.8	52.7	61.0
	SCtrack	73.5	66.8	68.0	68.7	**65.0**	**56.5**	**50.4**	**57.6**	**62.7**

Evaluation metrics. We use **Success** and **Precision** defined in one pass evaluation [37] as the evaluation metrics for 3D single object tracking. Specifically, **Success** is measured by the Intersection over Union (IoU) between predicted and ground truth 3D bounding boxes (BBox). **Precision** is measured by the area under the curve (AUC) for the distance between the centers of the two boxes, ranging from 0 to 2 meters. Details are in supplementary materials.

4.2 Comparison with State-of-the-art Trackers

Results on KITTI & Waymo. We present a comprehensive comparison between the proposed methods and the previous State-Of-The-Art (SOTA) methods, including recent Siamese trackers like SyncTrack and MBPTrack on the KITTI dataset. As shown in Tab. 1, SCtrack exhibits superior performance across various categories, achieving the highest mean Success and Precision rates of 72.0% and 89.1%, respectively. Moreover, SCtrack outperforms the previous leading method, *i.e.*, MBPTrack, by 1.7% while demonstrating notable advantages in running speed and architectural simplicity. Our method has a much simpler model structure with high performance, unlike the complex model design in the recent Siamese-based tracker, MBPTrack. This showcases the potential and effectiveness of our framework, which utilizes a joint appearance matching and motion modeling pipeline to accurately compute the target's relative motion between two consecutive frames.

In order to test the effectiveness of our proposed methods on different datasets, we evaluate the Car and Pedestrian models trained on the KITTI dataset using the Waymo dataset. The results, shown in Table 2, indicate that our SCtrack outperforms other comparison methods, especially in the Pedestrian category.

Table 3: The performance of different methods on the Nuscenes. "Mean" denotes the average results of four categories. Performance is arranged as "Success/Precision"

	Method	Success/Precision						
	Category	Car	Pedestrian	Truck	Trailer	Bus	Mean	Mean by Category
	Frame Num.	64159	33227	13587	3352	2953	117278	
Nuscenes	SC3D [23]	22.31/21.93	11.29/12.65	30.67/27.73	35.28/28.12	29.35/24.08	20.70/20.20	25.78/22.90
	P2B [64]	38.81/43.18	28.39/52.24	42.95/41.59	48.96/40.05	32.95/27.41	36.48/45.08	38.41/40.90
	PTT [67]	41.22/45.26	19.33/32.03	50.23/48.56	51.70/46.50	39.40/36.70	36.33/41.72	40.38/41.81
	BAT [95]	40.73/43.29	28.83/53.32	45.34/42.58	52.59/44.89	35.44/28.01	38.10/45.71	40.59/42.42
	PTTR [97]	51.89/58.61	29.90/45.09	45.30/44.74	45.87/38.36	43.14/37.74	44.50/52.07	43.22/44.91
	GLT-T [55]	48.52/54.29	31.74/56.49	52.74/51.43	57.60/52.01	44.55/40.69	44.42/54.33	47.03/50.98
	MBPTrack [87]	62.47/70.41	45.32/74.03	62.18/63.31	65.14/61.33	55.41/51.76	57.48/69.88	58.10/64.19
	SCtrack	**65.21/73.14**	**46.52/75.30**	**65.10/66.20**	**70.62/66.91**	**59.45/576.80**	**59.97/72.25**	**61.60/67.89**

This implies that our proposed framework has a strong ability to generalize to new scenes and datasets that it has not encountered before.

Results on NuScenes. We further conduct experiments on the challenging NuScenes dataset to validate the effectiveness of SCtrack. The following SOTA methods, which mostly follow the Siamese tracking and motion-centric paradigms, are compared to our method: SC3D [23], P2B [64], PTT [67], BAT [95], PTTR [97], M2Track [96], GLT-T [55] and MBPTrack [87]. Our SCtrack performs better than other comparison methods across all categories, as shown in Tab. 3. It is worth noting that SCtrack performs significantly better than other models in the categories of Trailer, and Bus, outperforming recent MBPTrack by 5.48% and 4.04% Success rate, respectively. These findings suggest that our proposed SCtrack framework can achieve outstanding results even with limited training data. The results indicate that our SCtrack has a strong generalization ability to new datasets, which can handle complex scenarios and hard cases.

Speed analysis. SCtrack with default hyper-parameters runs at a high speed of 60 FPS on a single RTX3090 GPU. The speed of SCtrack outperforms the recent SOTA Siamese tracker MBPtrack (50FPS), motion-centric tracker M2track (57FPS). It mainly attributes to that SCtrack has a much simpler model pipeline, which can exploit both appearance and motion clues within a single backbone network.

4.3 Ablation Studies

Analysis of spatially structured representation. To analyze the impact of the proposed spatially structured representation on tracking performance, we conduct an ablation study on SCtrack using the KITTI dataset. As shown in Tab. 4, the case (①) of using spatially structured embedding outperforms that of using point-wise embeddings (④) by a great margin in all four categories. Moreover, a similar phenomenon is observed even with direct concatenation between template and search frames. It fully validates that the spatially structured embedding is superior to the widely adopted point-wise feature embedding for point cloud tracking. This is attributed to the rich motion clues in spatially structured embedding space.

Analysis of data augmentation. To demonstrate the effectiveness of our data augmentation approach for 3D single object tracking, we train the SCtrack

Table 4: Ablation studies on our joint matching and motion modeling backbone on KITTI. “Concat" denotes channel concatenation for the template and search features. “S.S.Embed." and “Space-corr." denote spatially structured feature embedding and space correlation, while “Point-wise" denotes the features extracted by PointNet++. Performance is reported as Suc/Prec.

#Num	feature	Space-corr.	Concat	Data.Aug	Car	Pedestrian	Van	Cyclist
①	S.S.Embed.	✗	✓	✓	69.5/79.4	60.3/85.8	64.5/79.6	75.1/95.4
②	S.S.Embed.	✓	✗	✗	70.2/81.5	62.1/87.3	66.8/81.8	75.3/95.5
③	S.S.Embed.	✓	✗	✓	73.3/84.9	64.8/90.3	68.9/83.8	75.4/94.3
④	Point-wise	✗	✓	✓	65.7/77.6	52.3/81.5	63.7/77.8	75.8/94.8
⑤	Point-wise	✓	✗	✓	69.4/80.1	55.6/85.1	66.3/80.2	76.4/96.2

Table 5: Architectural variants of the structured embedding module in the KITTI dataset. We only report the mean value of all categories in Suc./Prec.

#Num	Channle/Layer	Speed	Mean
①	128/1	61 FPS	70.2/87.5
①	128/2	45 FPS	70.8/88.2
②	256/1	40 FPS	70.4/87.8
③	512/1	32 FPS	70.1/87.6

Table 6: Analysis of the space-correlated transformer in the KITTI dataset. We only report the mean value of all categories in Suc./Prec.

#Num	Region size	Mean(Suc./Prec.)
①	fixed 3×3	70.1/87.3
②	fixed 5×5	70.8/88.1
③	fixed 7×7	70.2/87.5
④	learned	71.5/88.9

model using the data augmentation introduced in M2Track, *i.e.*, without Data. Aug in Tab. 4. After analyzing the results, we conclude that our approach is more effective and efficient for two main reasons. Firstly, generating the current search region with the previous target as its center helps us to replicate the test error. Secondly, by translating along the XYZ axes, we are able to replicate more detailed motion patterns.

Analysis of space-correlated transformer. We study the effects of the proposed space-correlated transformer (SCT) in Tab. 5 and Tab. 6. The architectural variants of SCT, including the embedding dimensions and layer numbers, have varying impacts on the efficiency and performance of the tracker. In Tab. 5, we observe a significant speed decrease when the channel dimension and layer number are scaling to 256/512 and 2, respectively. However, the performance gain is relatively small (around -0.1% to $+0.6\%$). To achieve a better trade-off between performance and speed, we set the channel dimension and layer number to 256 and 1, respectively. As shown in Tab. 6, our varied-size region scheme outperforms the best case 5×5 of the fixed region size by 0.7% in terms of success, validating the effectiveness of adaptive region correlation.

Analysis of robustness to sparsity and distractors. In real scenarios, LiDAR point clouds often contain distractors and may be sparse. Therefore, it is crucial to evaluate the tracker's resilience to sparse point clouds and distractors. Fig.5 demonstrates SCtrack's superior performance in sparse scenes compared to M2Track, particularly in extremely sparse scenes. This is mainly due to the complementary effects of the appearance clues in our joint modeling framework, in contrast to the motion-centric tracking paradigm. Additionally, our methods show increased resilience to intra-class distractors.

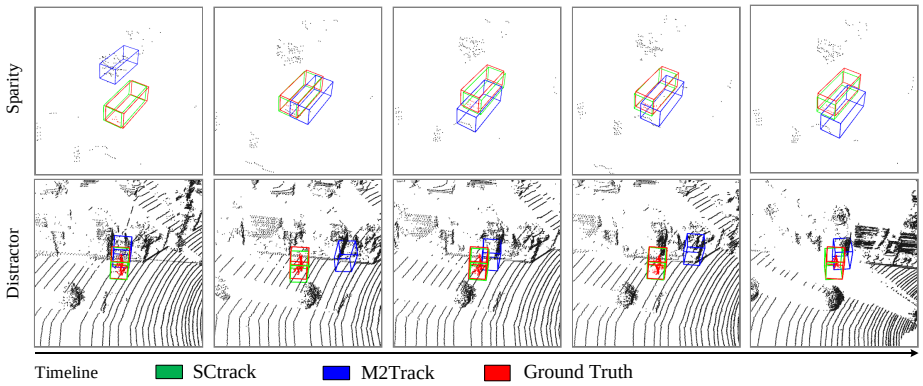

Fig. 5: Visualization of results on sparse and distracting point cloud scenarios in top and down row, respectively. It includes the motion-based tracker M2Track and our joint modeling method SCtrack.

5 Limitation and Discussion

Although our work has proven the effectiveness of structured embedding in 3D single object tracking task, we only evaluate it in our proposed tracker SCtrack. It will be our future work to validate the improvement of structured embedding on other Siamese-based trackers and other trackers with prediction styles. Compared to the motion-centric tracking paradigm, our joint appearance matching and motion modeling backbone is conceptually simple and consists of two modules, i.e., a structured embedding module and a space-correlated transformer. However, our tracking model could be improved by unifying these two modules by embedding the space correlation operator into the structured embedding stage, resulting in a more straightforward model pipeline.

6 Conclusion

This paper introduces SCtrack, a conceptually simple tracking framework. SCtrack provides 3D single object tracking area with a joint exploring appearance matching and motion modeling paradigm. It explores the spatially structured representation of point clouds instead of widely used point-wise representations in most 3D SOT methods. Moreover, in contrast to the complex matching-based fusion module, SCtrack uses a space-correlated transformer module to capture the appearance and motion information of targets in varying sizes and shapes. Extensive experiments demonstrate our SCtrack framework is efficient, achieving superior performance over all previous state-of-the-art trackers. We expect this work could attract more attention to the balance between leveraging appearance and motion clues on 3D single object tracking task. A deeper exploration of the joint backbone may leave as future work.

References

1. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.S.: Fully-convolutional siamese networks for object tracking. In: ECCVW (2016) 3
2. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.: Fully-convolutional Siamese networks for object tracking. In: ECCV (2016) 1
3. Bibi, A., Zhang, T., Ghanem, B.: 3D part-based sparse tracker with automatic synchronization and registration. In: CVPR (2016) 4
4. Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M.: Visual object tracking using adaptive correlation filters. In: CVPR (2010) 4
5. 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) 10
6. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with Transformers. In: ECCV (2020) 5
7. Chen, X., Yan, B., Zhu, J., Wang, D., Yang, X., Lu, H.: Transformer tracking. In: CVPR (2021) 3
8. Chen, X., Yan, B., Zhu, J., Wang, D., Yang, X., Lu, H.: Transformer tracking. In: CVPR (2021) 5
9. Chen, Z., Zhong, B., Li, G., Zhang, S., Ji, R.: Siamese box adaptive network for visual tracking. In: CVPR (2020) 3
10. Chu, X., Tian, Z., Zhang, B., Wang, X., Wei, X., Xia, H., Shen, C.: Conditional positional encodings for vision transformers. arXiv (2021) 8
11. Chu, X., Tian, Z., Zhang, B., Wang, X., Wei, X., Xia, H., Shen, C.: Conditional positional encodings for vision transformers. arXiv (2021) 9
12. Cui, Y., Fang, Z., Shan, J., Gu, Z., Sifan, Z.: 3D object tracking with Transformer. In: BMVC (2021) 2, 5, 10
13. Cui, Y., Jiang, C., Wang, L., Wu, G.: Mixformer: End-to-end tracking with iterative mixed attention. In: CVPR (2022) 3
14. 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) 4
15. Danelljan, M., Bhat, G., Shahbaz Khan, F., Felsberg, M.: ECO: Efficient convolution operators for tracking. In: CVPR (2017) 1
16. Danelljan, M., Hager, G., Shahbaz Khan, F., Felsberg, M.: Learning spatially regularized correlation filters for visual tracking. In: ICCV (2015) 4
17. Danelljan, M., Shahbaz Khan, F., Felsberg, M., Van de Weijer, J.: Adaptive color attributes for real-time visual tracking. In: CVPR (2014) 4
18. 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) 4
19. 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) 4, 8
20. Fan, H., Yang, Y., Kankanhalli, M.: Point 4D Transformer networks for spatio-temporal modeling in point cloud videos. In: CVPR (2021) 5
21. Fang, Z., Zhou, S., Cui, Y., Scherer, S.: 3D-SiamRPN: an end-to-end learning method for real-time 3D single object tracking using raw point cloud. IEEE Sensors Journal **21**(4), 4995–5011 (2020) 2

22. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the KITTI vision benchmark suite. In: CVPR (2012) 10
23. Giancola, S., Zarzar, J., Ghanem, B.: Leveraging shape completion for 3D Siamese tracking. In: CVPR (2019) 2, 4, 10, 12
24. Guo, D., Wang, J., Cui, Y., Wang, Z., Chen, S.: SiamCAR: Siamese fully convolutional classification and regression for visual tracking. In: CVPR (2020) 3
25. 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) 5
26. Guo, Q., Feng, W., Zhou, C., Huang, R., Wan, L., Wang, S.: Learning dynamic Siamese network for visual object tracking. In: ICCV (2017) 4
27. Held, D., Thrun, S., Savarese, S.: Learning to track at 100 fps with deep regression networks. In: ECCV (2016) 4
28. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: Exploiting the circulant structure of tracking-by-detection with kernels. In: ECCV (2012) 4
29. 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) 4
30. Hui, L., Wang, L., Cheng, M., Xie, J., Yang, J.: 3D Siamese voxel-to-BEV tracker for sparse point clouds. In: NeurIPS (2021) 2, 4, 10, 11
31. Hui, L., Wang, L., Tang, L., Lan, K., Xie, J., Yang, J.: 3d siamese transformer network for single object tracking on point clouds. In: European Conference on Computer Vision. pp. 293–310. Springer (2022) 2, 4, 5, 9, 10, 11
32. Hui, L., Yang, H., Cheng, M., Xie, J., Yang, J.: Pyramid point cloud Transformer for large-scale place recognition. In: ICCV (2021) 5
33. Ioffe, S., Szegedy, C.: Batch Normalization: Accelerating deep network training by reducing internal covariate shift. In: ICML (2015) 6
34. Kart, U., Kamarainen, J.K., Matas, J.: How to make an RGBD tracker? In: ECCV workshops (2018) 4
35. Kart, U., Lukezic, A., Kristan, M., Kämäräinen, J., Matas, J.: Object tracking by reconstruction with view-specific discriminative correlation filters. In: CVPR (2019) 4
36. 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) 1
37. 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) 1, 11
38. Lee, K.H., Hwang, J.N.: On-road pedestrian tracking across multiple driving recorders. IEEE Transactions on Multimedia **17**(9), 1429–1438 (2015) 1
39. Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J., Yan, J.: SiamRPN++: Evolution of siamese visual tracking with very deep networks. In: CVPR (2019) 3
40. Li, B., Yan, J., Wu, W., Zhu, Z., Hu, X.: High performance visual tracking with siamese region proposal network. In: CVPR (2018) 1, 3, 4
41. Li, J., Bian, S., Zeng, A., Wang, C., Pang, B., Liu, W., Lu, C.: Human pose regression with residual log-likelihood estimation. In: CVPR (2021) 9
42. 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) 4

43. Liu, R., Lehman, J., Molino, P., Such, F.P., Frank, E., Sergeev, A., Yosinski, J.: An intriguing failing of convolutional neural networks and the coordconv solution. In: NIPS (2018) 1
44. 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) 4
45. 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) 5
46. Luber, M., Spinello, L., Arras, K.O.: People tracking in RGB-D data with on-line boosted target models. In: IROS (2011) 4
47. 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) 4
48. Ma, T., Wang, M., Xiao, J., Wu, H., Liu, Y.: Synchronize feature extracting and matching: A single branch framework for 3d object tracking. In: CVPR. pp. 9953–9963 (2023) 4, 10
49. 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) 5
50. Meng, L., Li, H., Chen, B.C., Lan, S., Wu, Z., Jiang, Y.G., Lim, S.N.: Adavit: Adaptive vision transformers for efficient image recognition. In: CVPR. pp. 12309–12318 (2022) 8
51. Nie, J., Dong, Z., He, Z., Wu, H., Gao, M.: Faml-rt: Feature alignment-based multi-level similarity metric learning network for a two-stage robust tracker. Information Sciences (2023) 3
52. Nie, J., He, Z., Lv, X., Zhou, X., Chae, D.K., Xie, F.: Towards category unification of 3d single object tracking on point clouds. In: ICLR (2024) 4
53. Nie, J., He, Z., Yang, Y., Bao, Z., Gao, M., Zhang, J.: Osp2b: One-stage point-to-box network for 3d siamese tracking pp. 1285–1293 (2023) 4
54. Nie, J., He, Z., Yang, Y., Gao, M., Dong, Z.: Learning localization-aware target confidence for siamese visual tracking. IEEE Transactions on Multimedia (2022) 3
55. Nie, J., He, Z., Yang, Y., Gao, M., Zhang, J.: Glt-t: Global-local transformer voting for 3d single object tracking in point clouds. In: IJCAI (2023) 4, 5, 10, 12
56. Nie, J., He, Z., Yang, Y., Lv, X., Gao, M., Zhang, J.: Glt-t++: Global-local transformer for 3d siamese tracking with ranking loss. arXiv preprint arXiv:2304.00242 (2023) 4
57. Nie, J., Wu, H., He, Z., Gao, M., Dong, Z.: Spreading fine-grained prior knowledge for accurate tracking. IEEE Transactions on Circuits and Systems for Video Technology **32**(9), 6186–6199 (2022) 3
58. Pan, X., Xia, Z., Song, S., Li, L.E., Huang, G.: 3D object detection with pointformer. In: CVPR (2021) 5
59. Pang, Z., Li, Z., Wang, N.: Model-free vehicle tracking and state estimation in point cloud sequences. In: IROS (2021) 4, 10
60. 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) 4
61. Qi, C.R., Litany, O., He, K., Guibas, L.J.: Deep hough voting for 3D object detection in point clouds. In: ICCV (2019) 4
62. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: Deep learning on point sets for 3D classification and segmentation. In: CVPR (2017) 2, 3, 6

63. 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) 2, 3, 6
64. 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) 2, 4, 10, 11, 12
65. Rao, Y., Zhao, W., Liu, B., Lu, J., Zhou, J., Hsieh, C.J.: Dynamicvit: Efficient vision transformers with dynamic token sparsification. nips **34**, 13937–13949 (2021) 8
66. Real, E., Aggarwal, A., Huang, Y., Le, Q.V.: Regularized evolution for image classifier architecture search. In: AAAI (2019) 6
67. Shan, J., Zhou, S., Fang, Z., Cui, Y.: Ptt: Point-track-transformer module for 3d single object tracking in point clouds. In: IROS). IEEE (2021) 2, 4, 5, 10, 12
68. Spinello, L., Arras, K., Triebel, R., Siegwart, R.: A layered approach to people detection in 3D range data. In: AAAI (2010) 4
69. 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) 10
70. 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) 4
71. Tao, R., Gavves, E., Smeulders, A.W.: Siamese instance search for tracking. In: CVPR (2016) 4
72. 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) 4
73. Wang, Q., Gao, J., Xing, J., Zhang, M., Hu, W.: DCFNet: Discriminant correlation filters network for visual tracking. arXiv preprint arXiv:1704.04057 (2017) 4
74. 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) 5
75. 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) 8
76. 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) 2, 10
77. Wu, H., Nie, J., He, Z., Zhu, Z., Gao, M.: One-shot multiple object tracking in uav videos using task-specific fine-grained features. Remote Sensing **14**(16), 3853 (2022) 1
78. Xie, F., Chu, L., Li, J., Lu, Y., Ma, C.: Videotrack: Learning to track objects via video transformer. In: CVPR (2023) 3
79. Xie, F., Wang, C., Wang, G., Cao, Y., Yang, W., Zeng, W.: Correlation-aware deep tracking. In: CVPR (2022) 1, 3
80. Xie, F., Wang, C., Wang, G., Wankou, Y., Zeng, W.: Learning tracking representations via dual-branch fully transformer networks. In: ICCVW (2021) 3
81. Xie, F., Wang, Z., Ma, C.: Diffusiontrack: Point set diffusion model for visual object tracking. In: CVPR (2024) 3
82. Xie, F., Yang, W., Liu, B., Zhang, K., Wang, G., Zuo, W.: Learning spatio-appearance memory network for high-performance visual tracking. ICCVW (2021) 3

83. Xie, F., Yang, W., Wang, C., Chu, L., Cao, Y., Ma, C., Zeng, W.: Correlation-embedded transformer tracking: A single-branch framework. arXiv preprint arXiv:2401.12743 (2024) 3
84. Xu, A., Nie, J., He, Z., Lv, X.: Tm2b: Transformer-based motion-to-box network for 3d single object tracking on point clouds. IEEE Robotics and Automation Letters (2024) 2
85. Xu, B., Wang, N., Chen, T., Li, M.: Empirical evaluation of rectified activations in convolutional network. arXiv (2015) 8
86. Xu, T.X., Guo, Y.C., Lai, Y.K., Zhang, S.H.: Cxtrack: Improving 3d point cloud tracking with contextual information. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1084–1093 (2023) 4, 9, 10, 11
87. Xu, T.X., Guo, Y.C., Lai, Y.K., Zhang, S.H.: Mbptrack: Improving 3d point cloud tracking with memory networks and box priors. In: CVPR (2023) 4, 10, 11, 12
88. Xu, Y., Wang, Z., Li, Z., Yuan, Y., Yu, G.: SiamFC++: towards robust and accurate visual tracking with target estimation guidelines. In: AAAI (2020) 3
89. Yan, B., Peng, H., Fu, J., Wang, D., Lu, H.: Learning spatio-temporal transformer for visual tracking. In: ICCV (2021) 3
90. 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) 4
91. Ye, B., Chang, H., Ma, B., Shan, S.: Joint feature learning and relation modeling for tracking: A one-stream framework. arXiv (2022) 1, 3
92. 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) 4
93. Zhang, Q., Xu, Y., Zhang, J., Tao, D.: Vsa: Learning varied-size window attention in vision transformers. In: European conference on computer vision. pp. 466–483. Springer (2022) 5, 8
94. Zhang, Z., Peng, H.: Deeper and wider siamese networks for real-time visual tracking. In: CVPR (2019) 3
95. 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) 2, 4, 10, 11, 12
96. Zheng, C., Yan, X., Zhang, H., Wang, B., Cheng, S., Cui, S., Li, Z.: Beyond 3d siamese tracking: A motion-centric paradigm for 3d single object tracking in point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8111–8120 (2022) 2, 4, 9, 12
97. Zhou, C., Luo, Z., Luo, Y., Liu, T., Pan, L., Cai, Z., Zhao, H., Lu, S.: PTTR: Relational 3D point cloud object tracking with Transformer. arXiv (2021) 4, 5, 10, 12
98. Zhou, Y., Tuzel, O.: Voxelnet: End-to-end learning for point cloud based 3d object detection. In: CVPR. pp. 4490–4499 (2018) 6
99. 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) 5
100. Zhu, Z., Wang, Q., Li, B., Wu, W., Yan, J., Hu, W.: Distractor-aware Siamese networks for visual object tracking. In: ECCV (2018) 4
101. Zhu, Z., Nie, J., Wu, H., He, Z., Gao, M.: Msa-mot: Multi-stage association for 3d multimodality multi-object tracking. Sensors **22**(22), 8650 (2022) 1