MBPTrack: Improving 3D Point Cloud Tracking with Memory Networks and Box Priors

Tian-Xing Xu[1] Yuan-Chen Guo[1] Yu-Kun Lai[2] Song-Hai Zhang[1] *
[1] Tsinghua University, China [2] Cardiff University, United Kingdom
[1]{xutx21@mails., guoyc19@mails., shz@}tsinghua.edu.cn [2]LaiY4@cardiff.ac.uk

Abstract

*3D single object tracking has been a crucial problem for decades with numerous applications such as autonomous driving. Despite its wide-ranging use, this task remains challenging due to the significant appearance variation caused by occlusion and size differences among tracked targets. To address these issues, we present **MBPTrack**, which adopts a **M**emory mechanism to utilize past information and formulates localization in a coarse-to-fine scheme using **B**ox **P**riors given in the first frame. Specifically, past frames with targetness masks serve as an external memory, and a transformer-based module propagates tracked target cues from the memory to the current frame. To precisely localize objects of all sizes, MBPTrack first predicts the target center via Hough voting. By leveraging box priors given in the first frame, we adaptively sample reference points around the target center that roughly cover the target of different sizes. Then, we obtain dense feature maps by aggregating point features into the reference points, where localization can be performed more effectively. Extensive experiments demonstrate that MBPTrack achieves state-of-the-art performance on KITTI, nuScenes and Waymo Open Dataset, while running at 50 FPS on a single RTX3090 GPU.*

1. Introduction

The ability to track objects in 3D space is essential for numerous applications, including robotics [2, 12], autonomous driving [33, 15], and surveillance systems [27]. Given the initial state of a specific object, the aim of 3D single object tracking (SOT) is to estimate the pose and position of the tracked target in each frame. Early approaches [25, 18, 21] rely heavily on RGB information, which often struggle to handle changing lighting conditions. Therefore, recent research works [8, 23, 10, 11, 35, 31] have focused on using point clouds to solve 3D object tracking for their unique advantages, such as accurate spatial information and robustness to illumination changes.

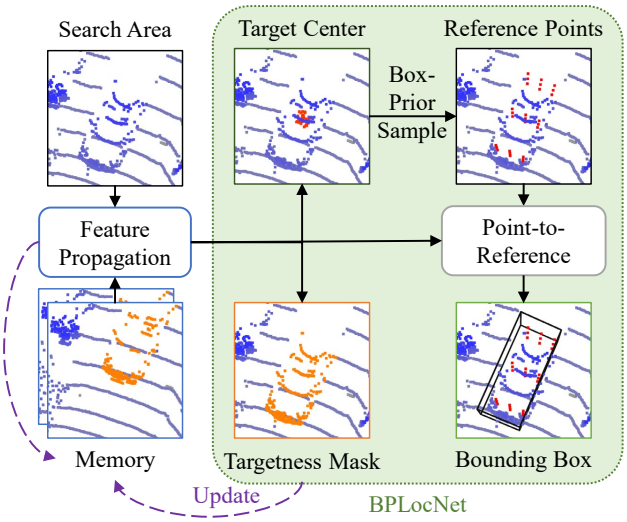

Figure 1: **An illustration of our proposed MBPTrack for 3D SOT task.** MBPTrack employs a memory mechanism to propagate target cues from historical frames and then utilizes a localization head, named BPLocNet, for coarse-to-fine bounding box prediction. BPLocNet first samples reference points around each predicted target center using a bounding box prior, which adaptively covers the tracked targets of all sizes. Then, BPLocNet aggregates local features from reference points for further refinement.

Existing methods [23, 30, 34, 24, 5, 10, 36, 11] for the 3D SOT task predominantly follow the Siamese paradigm, which takes the target template cropped from the previous frame and search area in the current frame as input, and then localizes the target in an end-to-end manner using a localization network such as Region Proposal Network [22] (RPN). Different from previous methods, M2-Track [35] explicitly models the target's motion between two successive frames and CXTrack [31] proposes to exploit the spatial contextual information across adjacent frames. Despite achieving promising results on popular datasets, the aforementioned methods propagate target cues solely from the

*corresponding author

latest frame to the current frame, thereby neglecting rich information contained in other past frames. This limitation renders 3D SOT a challenging task, especially in cases of large appearance variation or target disappearance caused by occlusion. To this end, TAT [16] exploits temporal information by sampling a set of high-quality target templates cropped from historic frames for reliable target-specific feature propagation. However, neglecting information in the latest frame could result in the network failing to capture lasting appearance changes, such as the gradual sparsification of point clouds as the tracked target moves further away. TAT also ignores the contextual information around the target, which is essential for 3D SOT [31], thereby leading to limited tracking performance.

In addition, the substantial differences in size and geometry across the various categories of tracked targets also pose challenges for 3D SOT, which has been overlooked by previous works. The localization networks adopted in existing methods can be categorized into two paradigms, namely point-based [36, 31, 23] and voxel-based [10]. For voxel-based localization heads like V2B [10], tracked targets with simple shapes and large sizes such as vehicles, can fit well in voxels, leading to more precise localization than point-based heads such as X-RPN [31]. However, for categories such as pedestrians, which have complex geometries and small sizes, voxelization leads to considerable information loss, thereby degrading tracking performance. As mentioned in V2B [10], the choice of different voxel sizes can significantly impact tracking performance.

To address the above issues, we present MBPTrack, a memory-based network for the 3D SOT task. Our approach relies on a memory mechanism to leverage rich spatial and temporal contextual information in historical frames and utilizes bounding box priors to address the challenge of size differences among tracked targets. Specifically, past frames with targetness masks serve as an external memory, and we draw inspiration from DeAOT [32], which has achieved great success in video object segmentation, to design a transformer-based module that propagates information from this memory to the current frame. It further decouples geometric features and targetness features into two processing branches with shared attention maps to enable effective learning of geometric information. Unlike TAT [16], MBPTrack fully utilizes both spatial and temporal contextual information around the target without cropping or sampling, thereby handling appearance variation and target disappearance/reappearance better than previous works. To achieve accurate localization of targets of different sizes, we introduce BPLocNet, a coarse-to-fine localization network that captures size information by leveraging the bounding box given in the first frame. BPLocNet first predicts the potential target centers as well as the targetness mask used to update the memory mechanism. We adopt a box-prior sampling method to sample reference points around the predicted target centers, adaptively covering the tracked target. Then, we aggregate point-wise features into the reference points, to obtain a dense feature map with spatial information, which is fed into a 3D CNN to predict precise bounding boxes. Extensive experiments demonstrate that MBPTrack outperforms existing methods by a large margin on three benchmark datasets, while running at 50 FPS on a single NVIDIA RTX3090 GPU. Furthermore, we demonstrate that using our proposed localization network in existing frameworks can consistently improve tracking accuracy.

In summary, our main contributions are as follows:

- To the best of our knowledge, we are the first to exploit both spatial and temporal contextual information in the 3D SOT task using a memory mechanism.
- We propose a localization network that utilizes box priors to localize targets of different sizes in a coarse-to-fine manner, which is shown to be effective in various 3D SOT frameworks.
- Experimental results demonstrate that MBPTrack outperforms existing methods, achieving state-of-the-art online tracking performance.

2. Related Work

As the pioneering work for point cloud-based 3D SOT, SC3D [8] computes feature similarity between the target template and a potentially large number of candidate proposals, which are sampled by Kalman filter in the search area. However, the heuristic sampling is time-consuming, and the pipeline cannot be end-to-end trained. To balance performance and efficiency, P2B [23] adopts a Region Proposal Network [22] to generate high-quality 3D proposals. The proposal with the highest score is selected as the final output. Many follow-up works adopt the same paradigm. MLVSNet [30] enhances P2B by performing multi-level Hough voting for effectively aggregating information at different levels. BAT [34] designs a box-aware feature fusion module to capture the explicit part-aware structure information. V2B [10] proposes to transform point features into a dense bird's eye view feature map to tackle the sparsity of point clouds. LTTR [5], PTTR [36], CMT [9] and STNet [11] introduce various attention mechanisms into the 3D SOT task for better target-specific feature propagation. PTTR [36] also proposes a light-weight Prediction Refinement Module for coarse-to-fine localization. However, these methods rely wholly on the appearance of the target, so tend to drift towards distractors in dense traffic scenes [35]. To this end, M2-Track [35] introduces a motion-centric paradigm that explicitly models the target's motion between two adjacent frames. CXTrack [31] exploits contextual information across adjacent frames to im-

prove tracking results. Although achieving promising results, these methods only exploit the target cues in the latest frame. The overlook of rich information in historical frames may hinder precise localization in the case of large appearance variation or target disappearance caused by occlusion.

TAT [16] is the first work to exploit the rich temporal information. It samples high-quality target templates from historical frames and adopts an RNN-based module [4] to aggregation target cues from multiple templates. However, the overlook of low-quality target templates in the latest frame makes the network fail to capture lasting appearance variation caused by long-term partial occlusion. It also ignores the spatial contextual information in the historical frames, which is essential for 3D SOT, as mentioned in CXTrack [31]. Besides, none of the aforementioned methods consider the size differences of tracked objects. For example, compared with pedestrian, vehicles have simple shapes and large sizes, which fit well in voxels. Thus voxel-based networks such as STNet [11] achieve better performance on the Car category than point-based networks like CXTrack [31], but face great challenges on the Pedestrian category. We argue that object occlusion and size difference are two main factors that pose great challenges for 3D SOT.

3. Method

3.1. Problem Definition

Given the 3D bounding box (BBox) of a specific target in the first frame, 3D SOT aims to localize the target by predicting its bounding box in subsequent frames. The frame at timestamp t is represented as a point cloud $\mathcal{P}_t \in \mathbb{R}^{\dot{N}_t \times 3}$, where $\dot{N}_t$ is the number of points. The 3D BBox $\mathcal{B}_t \in \mathbb{R}^7$ at timestamp t is parameterized by its center (xyz coordinates), orientation (heading angle θ around the up-axis) and size (width w, length l and height h). Even for non-rigid objects like pedestrians, the size of the tracked target remains approximately unchanged in 3D SOT. Thus, for each frame $\mathcal{P}_t$, we only regress the translation offset $(\Delta x_t, \Delta y_t, \Delta z_t)$ and the rotation angle $(\Delta\theta_t)$ from $\mathcal{P}_{t-1}$ to $\mathcal{P}_t$ to simplify the tracking task, with access to historical frames $\{\mathcal{P}_i\}_{i=1}^{t}$. The 3D BBox $\mathcal{B}_t$ can be easily obtained by applying a rigid body transformation to $\mathcal{B}_{t-1}$ from the previous frame. Additionally, to indicate a more precise location of the tracked target at timestamp t, we predict a targetness mask $\dot{\mathcal{M}}_t = (m_t^1, m_t^2, \cdots, m_t^{\dot{N}_t}) \in \mathbb{R}^{\dot{N}_t}$ frame by frame, where the mask m_t^i represents the possibility of the i-th point $p_t^i \in \mathcal{P}_t$ being within $\mathcal{B}_t$ ($\dot{\mathcal{M}}_1$ is computed using the given $\mathcal{B}_1$). Hence, we can formulate 3D SOT at timestamp $t (t > 1)$ as learning the following mapping

$$\mathcal{F}(\{\mathcal{P}_i\}_{i=1}^{t-1}, \{\dot{\mathcal{M}}_i\}_{i=1}^{t-1}, \mathcal{P}_t, \mathcal{B}_1) \mapsto (\Delta x_t, \Delta y_t, \Delta z_t, \Delta\theta_t, \dot{\mathcal{M}}_t) \quad (1)$$

3.2. Overview

Following Eq. 1, we design a memory-based framework, MBPTrack, to capture the spatial and temporal information in the historical frames and tackle the size difference across various categories of tracked targets. As illustrated in Fig. 2, given an input sequence $\{\mathcal{P}_i\}_{i=1}^{t}$ of a dynamic 3D scene, we first employ a shared backbone to embed the local geometric information in each frame into point features, denoted by $\mathcal{X}_i \in \mathbb{R}^{N \times C}$ for the i-th frame. Here N is the number of point features and C denotes the number of feature channels. The corresponding targetness masks $\mathcal{M}_i \in \mathbb{R}^{N \times 1} (i < t)$ are obtained from $\dot{\mathcal{M}}_i$ (either from the first frame or estimated from past frames) to identify the tracked target in past frames. The targetness mask $\mathcal{M}_t$ for the current frame is initialized with 0.5 as it is unknown. Then, we design a transformer-based decoupling feature propagation module (DeFPM, Sec. 3.3) to leverage both temporal and spatial context present in the dynamic 3D scene. Finally, we develop a simple yet efficient localization network, BPLocNet, which formulates the localization of targets as coarse-to-fine prediction using box priors to tackle size differences among tracked targets.

3.3. Decoupling Feature Propagation Module

Inspired by DeAOT [32] in video object segmentation, we introduce a decoupling feature propagation module (DeFPM) into the 3D SOT task, which relies on a memory mechanism to explore both spatial and temporal information from the past frames while propagating target cues into the current frame. DeAOT [32] indicates that integrating targetness information will inevitably cause the loss of object-agnostic geometric information. Hence, DeFPM decouples the propagation of geometric features and mask features to learn more distinct geometric embeddings, which is essential for handling sparse point clouds. DeFPM consists of $N_L = 2$ identical layers with two parallel branches, as illustrated in Fig. 3. Each layer includes three main parts, *i.e.*, a cross-attention module that propagates both target cues and temporal context from past frames to the current frame, a self-attention module that captures long-range contextual information in the current frame, and a feed-forward network for feature refinement.

To formulate the feature propagation from the memory to the current frame, we first define the input of the l-th layer. Let $X^{(l-1)} \in \mathbb{R}^{N \times C}$ and $X_{\mathrm{m}}^{(l-1)} \in \mathbb{R}^{TN \times C}$ denote the geometric features in the current frame and from the memory, where T represents the memory size (the number of memory frames). We adopt a "pre-norm" transformer design [19], which employs a layer normalization [1] oper-

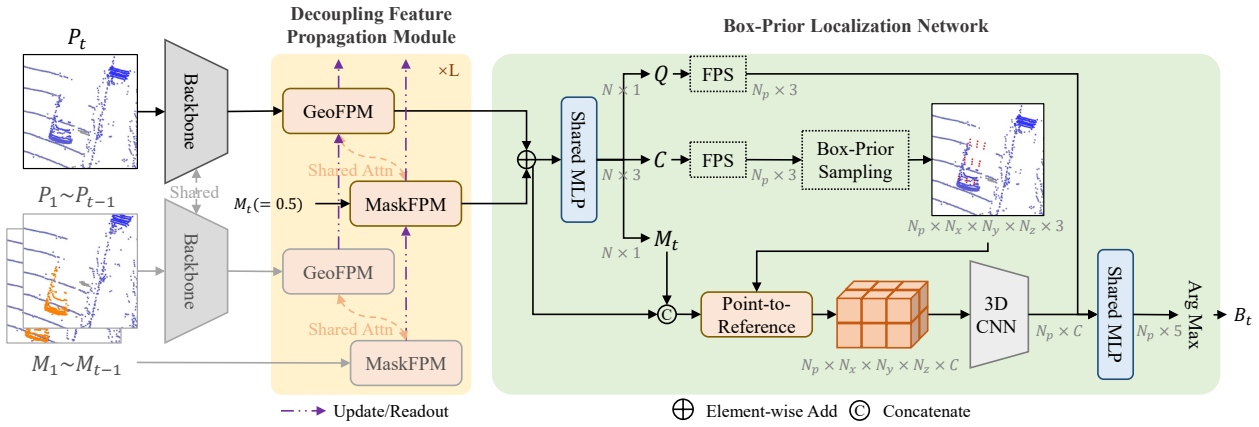

Figure 2: **An overview of our proposed MBPTrack architecture.** We employ a backbone to extract geometric features. Then, the past frames with their targetness mask serve as an external memory and the decoupling feature propagation module is used to propagate rich target cues from historical frames. We also propose a box-prior localization network, which leverages box priors to sample reference points that adaptively cover the target of different sizes for precise localization.

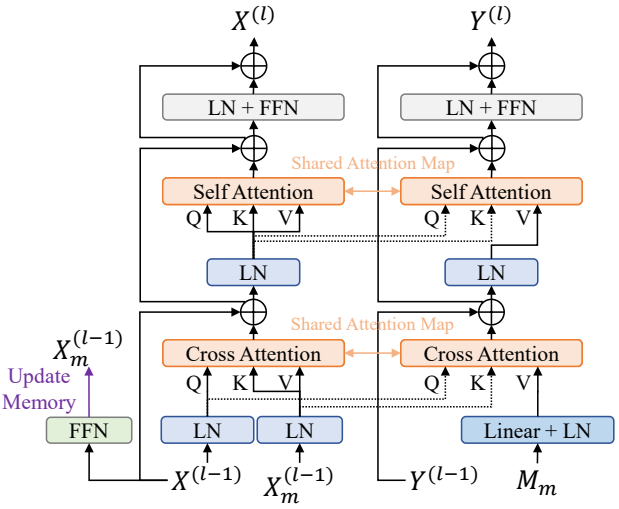

Figure 3: **Decoupling Feature Propagation Module (DeFPM).** DeFPM decouples the propagation of geometric information and targetness information into two branches to avoid the loss of geometric information into deep propagation layers. Both branches have the same hierarchical structure with shared attention maps.

ation $\mathrm{LN}(\cdot)$ before the attention mechanism, written as

$$\overline{X} = \mathrm{LN}(X^{(l-1)}) \tag{2}$$

$$\overline{X}_M = \mathrm{LN}(X_M^{(l-1)}) \tag{3}$$

The attention mechanism [28] is the basic block of our proposed DeFPM, which takes the query $Q \in \mathbb{R}^{n \times d}$, key $K \in \mathbb{R}^{n \times d}$ and value $V \in \mathbb{R}^{n \times d}$ as input, and then computes the similarity matrix between the query and the key to obtain a weighted sum of V

$$\mathrm{Attn}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d}})V. \tag{4}$$

Notably, we add positional embeddings of the coordinates to the query and key, which is omitted from the formula of attention for brevity. Hence, the cross attention operation of the geometric branch can be formulated as

$$\widetilde{X} = X^{(l-1)} + \mathrm{Attn}(\overline{X}W_Q^c, \overline{X}_M W_K^c, \overline{X}_M W_{V,G}^c), \tag{5}$$

where $W_Q^c \in \mathbb{R}^{C \times d}$, $W_K^c \in \mathbb{R}^{C \times d}$ and $W_{V,G}^c \in \mathbb{R}^{C \times d}$ are learnable parameter matrices and d is the channel dimension. To update the memory bank, we adopt a lightweight feed-forward network on the input $X^{(l-1)}$ to obtain the reference features of the current frame, which ensures the effectiveness and efficiency of the memory mechanism.

The mask branch is designed to propagate targetness information into the current frame to indicate the tracked target, which shares the attention maps with the geometric branch. Suppose $Y^{(l-1)} \in \mathbb{R}^{N \times C}$ denotes the mask features output by the $(l-1)$-th layer ($Y^{(0)} = \phi(\mathcal{M}_t)$, where ϕ is a linear projection layer) and $M_m \in \mathbb{R}^{TN \times 1}$ denotes the targetness masks of all the frames saved in memory. We project the input masks M_m to mask embeddings using a shared linear transformation $\varphi(\cdot)$ with a layer normalization

$$\overline{Y_m} = \mathrm{LN}(\varphi(M_m)) \tag{6}$$

The output of the cross-attention operation is given by

$$\widetilde{Y} = Y^{(l-1)} + \mathrm{Attn}(\overline{X}W_Q^c, \overline{X}_M W_K^c, \overline{Y}_M W_{V,M}^c), \tag{7}$$

To explore the contextual information within the current frame and enhance the point features, DeFPM subsequently

employs a global self-attention operation, which can be formulated similarly to the cross-attention operation

$$\widehat{X} = \widetilde{X} + \text{Attn}(\dot{X}W^s_Q, \dot{X}W^s_K, \dot{X}W^s_{V,G}) \quad (8)$$

$$\widehat{Y} = \widetilde{Y} + \text{Attn}(\dot{X}W^s_Q, \dot{X}W^s_K, \dot{Y}W^s_{V,G}) \quad (9)$$

$$\text{where} \quad \dot{X} = \text{LN}(\widetilde{X}), \dot{Y} = \text{LN}(\widetilde{Y}) \quad (10)$$

Finally, two fully connected feed-forward networks are used to separately refine the point features and mask features, which can be written as

$$X^{(l)} = \widehat{X} + \text{FFN}(\text{LN}(\widehat{X})) \quad (11)$$

$$Y^{(l)} = \widehat{Y} + \text{FFN}(\text{LN}(\widehat{Y})) \quad (12)$$

$$\text{where} \quad \text{FFN}(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2 \quad (13)$$

3.4. Box-Prior Localization Network

The difference in size and geometry among the tracked targets poses great challenges to existing localization networks. To address the above concern, we design a novel localization network, named BPLocNet, that formulates the localization of the target in a coarse-to-fine manner using box priors given in the first frame, as illustrated in Fig. 2. Previous works [23, 36, 10, 11] mainly use the bounding box to crop the target template from previous frames while ignoring the size information about the target. Hence, we propose to adaptively sample reference points that roughly cover the targets of different sizes using the given bounding box, and then refine the prediction for precise localization.

Box-prior sampling. We apply a shared MLP on the fused point features $F = X^{(N_L)} + Y^{(N_L)} \in \mathbb{R}^{N\times C}$ to predict the potential target center $\mathcal{C} \in \mathbb{R}^{N\times 3}$ via Hough voting, as well as a point-wise targetness mask $\mathcal{M}_t$ for memory update. Each target center prediction can be viewed as a proposal center, while we use further point sampling to sample a subset $\mathcal{C}_p$ in $\mathcal{C}$ to be of size N_p for efficiency. Suppose w, l, h denote the width, length and height of the 3D bounding box $\mathcal{B}_1$ given in the first frame along each axis. Leveraging the proposal centers $\mathcal{C}_p$ and the size information w, l, h, we can sample a set of reference points $\mathcal{R}_c$ for each center $c \in \mathcal{C}_p$ (as shown in Fig. 4), which can be formulated as follows

$$\mathcal{R}_c = \left\{ c + s_{i,j,k} \middle| s_{i,j,k} = \left(\frac{2i - n_x - 1}{2n_x} w, \frac{2j - n_y - 1}{2n_y} l, \frac{2k - n_z - 1}{2n_z} h \right) \ \forall i \in [1, n_x], j \in [1, n_y], k \in [1, n_z] \right\} \quad (14)$$

where n_x, n_y, n_z denote the numbers of samples along axes. All reference points form a point set $\mathcal{R} = \bigcup_{c\in\mathcal{C}_p} \mathcal{R}_c$. Despite the simplicity, reference points have the following properties, which can benefit the localization task:

- **Coarse prediction.** We observe the rotation angles $\Delta\theta$ of tracked targets between two consecutive frames are small in most cases, especially for vehicle tracking. Hence, reference points can serve as a coarse prediction of the localization of targets.

- **Adaptive to different sizes.** By leveraging size information, reference points are uniformly distributed in the bounding box of tracked targets, making the network adaptive to targets of different sizes.

- **Shape prior.** For targets such as vehicles that have simple shapes, reference points provide a strong shape prior, which roughly cover the targets in 3D space.

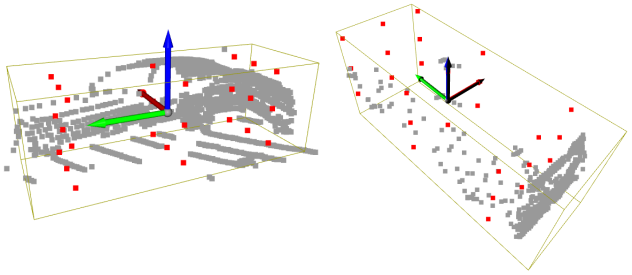

Figure 4: **Box-Prior Sampling.** Red indicates the reference points that roughly cover the tracked targets. We sample 3 points along each axis for visualization.

Point-to-reference feature transformation. Due to the fixed relative position of reference points, we can obtain a 3D dense feature map from unordered point features, where the localization can be performed more effectively. We first integrate the targetness mask score $m_j \in \mathcal{M}_t$ into the point features $f_j \in F$ using a shared MLP $h(\cdot)$

$$\widehat{f_j} = h([f_j; m_j]) \quad (15)$$

where $[\cdot;\cdot]$ is the concatenation operation. Then, we adopt a modified EdgeConv [29] operator to aggregate information from the neighborhood points $j \in \mathcal{N}(r)$ with features $\widehat{f_j} \in F$ to the reference point $r \in \mathcal{R}$, written as follows

$$f_r = \max_{j:j\in\mathcal{N}(r)} e([\widehat{f_j}; x_j - r; r]) \quad (16)$$

where $e(\cdot)$ denotes a shared MLP. We arrange the features f_r in a predefined order with respect to the coordinates of reference points to generate the 3D dense feature map $\mathcal{Z} \in \mathbb{R}^{n_x\times n_y\times n_z\times C}$ for each proposal $c \in \mathcal{C}$. Finally, the 3D feature maps $\mathcal{Z}$ are fed into a shared 3D CNN to obtain proposal-wise features.

Coarse-to-fine score prediction. As mentioned in M2-Track [35], distractors are widespread in dense traffic scenes. 3D CNN mainly captures the appearance of the target, while failing to distinguish the target from distractors in dense traffic scenes. Thus, we also use the point features

output by DeFPM to predict a quality score $\mathcal{Q} \in \mathbb{R}^{N\times 1}$, responsible for measuring the distance between the predicted target center $\mathcal{C}$ and the ground truth. We project the quality score $\mathcal{Q}$ to quality embeddings using a shared linear transformation, and then add them to the proposal-wise features to obtain the proposal box parameters $\mathcal{B}^{N_p\times 4}$ with refined targetness scores $S \in \mathbb{R}^{N_p\times 1}$.

3.5. Implementation Details

Loss functions. The predicted targetness mask $\mathcal{M}_t$ is supervised by a standard cross-entropy loss, denoted as $\mathcal{L}_{\text{m}}$. For the target center prediction, we use an MSE (Mean Squared Error) loss $\mathcal{L}_{\text{c}}$. Following P2B [23], we consider predicted centers near the ground truth target center (<0.3m) as positive and others as negative to obtain the ground truth of quality score Q and targetness score S, which are supervised by cross-entropy loss $\mathcal{L}_{\text{q}}$ and $\mathcal{L}_{\text{s}}$, respectively. Only the bounding box parameters of positive proposals are supervised via a smooth-L1 loss $\mathcal{L}_{\text{bbox}}$. The final loss of each frame can be written as

$$\mathcal{L} = \lambda_{\text{m}}\mathcal{L}_{\text{m}} + \lambda_{\text{c}}\mathcal{L}_{c} + \lambda_{\text{q}}\mathcal{L}_{\text{q}} + \lambda_{\text{s}}\mathcal{L}_{s} + \mathcal{L}_{\text{bbox}} \tag{17}$$

where λ_{m}(=0.2), λ_{c}(=10.0), λ_{q}(=1.0) and λ_{s}(=1.0) hyperparameters are used to balance the component losses.

Positive sampling. We observe that for objects with complex shapes such as pedestrians, it is difficult to regress precise target centers for all point features. Hence, positive proposals (<0.3m) for box parameter prediction are much fewer than negative proposals. To balance positive and negative proposals, we replace part of the predicted proposal centers with positive centers generated by applying a small perturbation to the ground truth center for nonrigid objects such as pedestrians and cyclists during training.

Training & Testing. We sample clips from the whole sequence to form training samples. The length of a clip is set to 8. We sum the losses of frames to obtain the loss of each clip. To balance efficiency and effectiveness, the memory size of MBPTrack is set to 2 for training and 3 for testing (Sec. 4.3). MBPTrack reuses the previous prediction if the tracked target is lost ($\max(\mathcal{M}_t) < 0.2$). More details can be seen in the supplementary material.

4. Experiments

4.1. Settings

Datasets. We adopt three popular large-scale datasets, namely KITTI [7], NuScenes [3] and Waymo Open Dataset [26] (WOD), to validate the effectiveness of our model. KITTI contains 21 video sequences for training and 29 video sequences for testing. Due to the inacessibility of the test labels, we follow previous work [23] and split the training dataset into three subsets, sequences 0-16 for training, 17-18 for validation, and 19-20 for testing. NuScenes is more challenging than KITTI for its larger data volumes, containing 700/150/150 scenes for training/validation/testing. For WOD, we follow LiDAR-SOT [20] to evaluate our method on 1121 tracklets, which are divided into easy, medium and hard subsets based on the sparsity of point clouds.

Evaluation metrics. We follow One Pass Evaluation [14]. For the predicted and ground truth bounding boxes, Success is defined as the Area Under Curve (AUC) for the plot showing the ratio of frames where the Intersection Over Union (IOU) is greater than a threshold, ranging from 0 to 1, while Precision denotes the AUC for the plot showing the ratio of frames where the distance between their centers is within a threshold, ranging from 0 to 2 meters.

Table 1: **Comparisons with the state-of-the-art methods on KITTI dataset**. "Mean" is the average result weighted by frame numbers. "Underline" and "**Bold**" denote previous and current best performance, respectively. Success/Precision are used for evaluation.

Method	Car (6424)	Pedestrian (6088)	Van (1248)	Cyclist (308)	Mean (14068)
SC3D	41.3/57.9	18.2/37.8	40.4/47.0	41.5/70.4	31.2/48.5
P2B	56.2/72.8	28.7/49.6	40.8/48.4	32.1/44.7	42.4/60.0
3DSiamRPN	58.2/76.2	35.2/56.2	45.7/52.9	36.2/49.0	46.7/64.9
LTTR	65.0/77.1	33.2/56.8	35.8/45.6	66.2/89.9	48.7/65.8
MLVSNet	56.0/74.0	34.1/61.1	52.0/61.4	34.3/44.5	45.7/66.7
BAT	60.5/77.7	42.1/70.1	52.4/67.0	33.7/45.4	51.2/72.8
PTT	67.8/81.8	44.9/72.0	43.6/52.5	37.2/47.3	55.1/74.2
V2B	70.5/81.3	48.3/73.5	50.1/58.0	40.8/49.7	58.4/75.2
CMT	70.5/81.9	49.1/75.5	54.1/64.1	55.1/82.4	59.4/77.6
PTTR	65.2/77.4	50.9/81.6	52.5/61.8	65.1/90.5	57.9/78.1
STNet	72.1/84.0	49.9/77.2	58.0/70.6	73.5/93.7	61.3/80.1
TAT	72.2/83.3	57.4/84.4	58.9/69.2	74.2/93.9	64.7/82.8
M2-Track	65.5/80.8	61.5/88.2	53.8/70.7	73.2/93.5	62.9/83.4
CXTrack	69.1/81.6	67.0/91.5	60.0/71.8	74.2/**94.3**	67.5/85.3
MBPTrack	**73.4/84.8**	**68.6/93.9**	**61.3/72.7**	**76.7/94.3**	**70.3/87.9**
Improvement	↑1.2/↑0.8	↑1.6/↑2.4	↑1.3/↑0.9	↑2.5/0.0	↑2.8/↑2.6

4.2. Results

We present a comprehensive comparison of our method with the previous state-of-the-art approaches, namely SC3D [8], P2B [23], 3DSiamRPN [6], LTTR [5], MLVSNet [30], BAT [34], PTT [24], V2B [10], CMT [9], PTTR [36], STNet [11], TAT [16], M2-Track [35] and CXTrack [31] on the KITTI dataset. The published results from corresponding papers are reported. As illustrated in Tab. 1, MBPTrack surpasses other methods on all categories, with an obvious improvement in average Success and Precision. Notably, compared with point-based methods such as CXTrack or M2Track, methods using voxel-based localization heads like STNet and V2B achieve satisfying results on the Car category. We presume that the improvement stems from the simple shape and large size of cars, which fit well in voxels. However, STNet and V2B perform poorly on the

Table 2: **Comparison with state of the arts on Waymo Open Dataset.**

Method	Vehicle(185731)				Pedestrian(241752)				Mean(427483)
	Easy	Medium	Hard	Mean	Easy	Medium	Hard	Mean	
P2B	57.1/65.4	52.0/60.7	47.9/58.5	52.6/61.7	18.1/30.8	17.8/30.0	17.7/29.3	17.9/30.1	33.0/43.8
BAT	61.0/68.3	53.3/60.9	48.9/57.8	54.7/62.7	19.3/32.6	17.8/29.8	17.2/28.3	18.2/30.3	34.1/44.4
V2B	64.5/71.5	55.1/63.2	52.0/62.0	57.6/65.9	27.9/43.9	22.5/36.2	20.1/33.1	23.7/37.9	38.4/50.1
STNet	65.9/72.7	57.5/66.0	54.6/64.7	59.7/68.0	29.2/45.3	24.7/38.2	22.2/35.8	25.5/39.9	40.4/52.1
TAT	66.0/72.6	56.6/64.2	52.9/62.5	58.9/66.7	32.1/49.5	25.6/40.3	21.8/35.9	26.7/42.2	40.7/52.8
CXTrack	63.9/71.1	54.2/62.7	52.1/63.7	57.1/66.1	35.4/55.3	29.7/47.9	26.3/44.4	30.7/49.4	42.2/56.7
M2Track	68.1/75.3	**58.6**/66.6	55.4/64.9	61.1/69.3	35.5/54.2	30.7/48.4	29.3/45.9	32.0/49.7	44.6/58.2
MBPTrack	**68.5/77.1**	58.4/**68.1**	**57.6/69.7**	**61.9/71.9**	**37.5/57.0**	**33.0/51.9**	**30.0/48.8**	**33.7/52.7**	**46.0/61.0**
Improvement	↑0.4/↑1.8	↓0.2/↑1.5	↑2.2/↑4.8	↑0.8/↑2.6	↑2.0/↑1.7	↑2.3/↑3.5	↑0.7/↑2.9	↑1.7/↑3.0	↑1.4/↑2.8

Table 3: **Comparisons with the state-of-the-art methods on NuScenes dataset**.

Method	Car(64159)	Pedestrian(33227)	Truck(13587)	Trailer(3352)	Bus(2953)	Mean(117278)
SC3D	22.31/21.93	11.29/12.65	30.67/27.73	35.28/28.12	29.35/24.08	20.70/20.20
P2B	38.81/43.18	28.39/52.24	42.95/41.59	48.96/40.05	32.95/27.41	36.48/45.08
BAT	40.73/43.29	28.83/53.32	45.34/42.58	52.59/44.89	35.44/28.01	38.10/45.71
M2-Track	55.85/65.09	32.10/60.92	57.36/59.54	57.61/58.26	51.39/51.44	49.23/62.73
MBPTrack	**62.47/70.41**	**45.32/74.03**	**62.18/63.31**	**65.14/61.33**	**55.41/51.76**	**57.48/69.88**
Improvement	↑6.62/↑5.32	↑13.22/↑13.11	↑4.82/↑3.77	↑7.53/↑3.07	↑4.02/↑0.32	↑8.25/↑7.15

Pedestrian category, which has small size and complex geometry. Voxelization results in inevitable information loss, causing the network to fail to distinguish the target from distractors. Leveraging box priors and a memory mechanism, our method achieves state-of-the-art performance on both categories. Compared with TAT, which samples high-quality target templates from historical frames, our method obtains consistent performance gains across all categories. It indicates that our method benefits a lot from spatial and temporal information that TAT discards during sampling.

For further explanation, we present a visual analysis of the tracking results on KITTI. As shown in Fig. 5, CXTrack [31], which adopts a point-based head, fails to predict the orientation accurately on the Car category, while the predicted bounding boxes by our method hold tight to the ground truths. For pedestrians, all methods tend to drift towards intra-class distractors due to the large appearance variation caused by heavy occlusion. However, only MBPTrack can accurately track the target after the occlusion disappears, owing to the sufficient use of temporal information.

We also evaluate the KITTI pretrained models on WOD [26], following previous work [11]. The corresponding categories between KITTI and WOD datasets are Car→Vehicle and Pedestrian→Pedestrian. The experimental results, as presented in Tab. 2, indicate that MBPTrack yields competitive or better tracking results than other methods under different levels of sparsity. In conclusion, our proposed method not only precisely tracks targets of all sizes but also generalizes well to unseen scenarios.

Table 4: **Model complexity and inference time.**

Component	FLOPs	#Params	Inference Speed
backbone	1.59G	1.19M	4.6ms
DeFPM	0.22G	2.67M	7.9ms
BPLocNet	1.07G	3.52M	3.6ms
pre/post-process	-	-	3.9ms
MBPTrack	2.88G	7.38M	20.0ms (50FPS)

NuScenes [3] presents a greater challenge for 3D SOT task than KITTI due to its larger data volumes and lower frequency for annotated frames (2Hz for NuScenes v.s. 10Hz for KITTI and WOD). We conduct a comparison of our approach with previous methods on the NuScenes dataset following M2-Track [35]. As shown in Tab. 3, our method achieves a consistent and large performance gain compared with the previous state-of-the-art method, M2-Track. Leveraging the rich temporal and spatial information contained in the historical frames, MBPTrack exhibits superior performance over methods that only consider two frames when large appearance variation occurs between them.

Fig. 4 shows the model complexity and average inference time of different components in the Car category on KITTI. Our experiments are conducted on a single NVIDIA RTX 3090. MBPTrack achieves 50 FPS, with 4.6ms for feature extraction, 7.9ms for feature propagation, 3.6ms for localization and 3.9ms for pre/post-processing. Using a more light-weight attention mechanism (e.g. [17, 13]) in DeFPM may further increase the running speed.

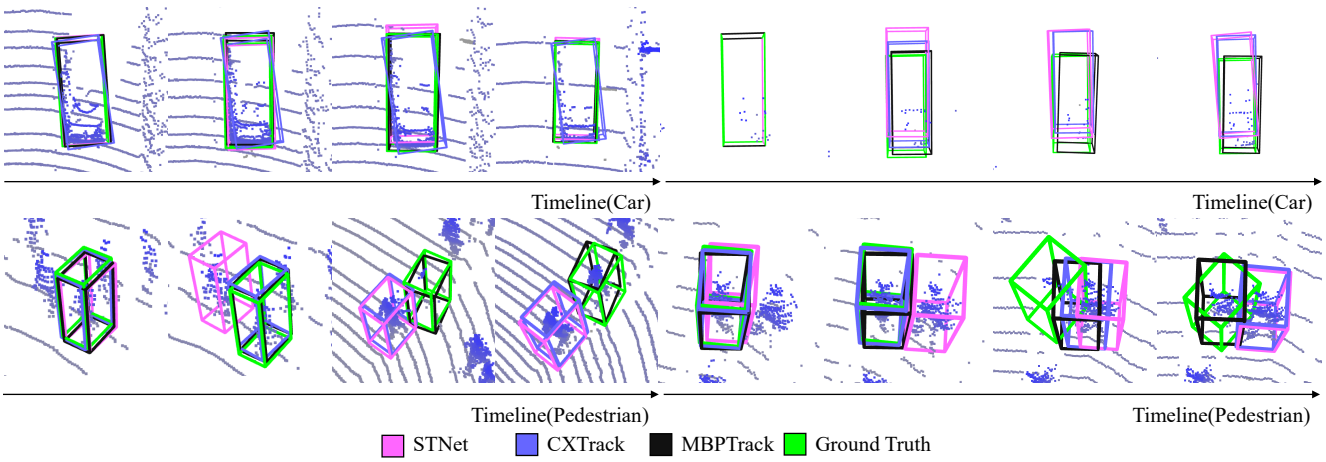

Figure 5: **Visualization of tracking results compared with state-of-the-art methods.**

Table 5: **Ablation study of the memory size.**

#Frames	Car	Pedestrian	Van	Cyclist	Mean
1	72.5/83.7	63.2/88.9	**62.8/74.3**	74.6/93.7	67.7/85.3
2	72.5/82.8	66.8/91.9	62.0/73.4	76.6/94.4	69.2/86.2
3 (Ours)	73.4/84.8	**68.6/93.9**	61.3/72.7	76.7/94.3	**70.3/87.9**
4	**74.9/86.7**	66.4/92.1	60.7/72.0	76.7/94.4	70.0/**87.9**
5	74.1/85.5	66.7/92.2	61.2/72.4	76.7/94.4	69.8/87.4
6	72.2/83.5	65.6/91.0	60.1/71.5	**77.4/94.6**	68.4/85.9

4.3. Ablation Studies

Memory Size. Memory size is defined as the number of historical frames with their corresponding targetness masks saved in the memory. To explore the impact of memory size, we conduct experiments on KITTI and report the results in Tab. 5. Notably, we train and test our model using only one previous frame when the memory size is set as 1. In this case, MBPTrack is degraded to a Siamese-based network, same as previous work [11, 31, 35]. Otherwise, we adopt the default training settings. Compared with Siamese-based version, our method benefits a lot from exploiting temporal information, leading to a significant improvement on the average metrics. This demonstrates the importance of historical information. We also observe that performance begins to decline when using more than 4 frames for tracking. Larger memory size can provide more reference information to handle sudden appearance variation caused by occlusion, but may fail to tackle lasting appearance variation, such as the gradual sparsification of point clouds as the tracking target moves further away. Besides, the memory size at which the model's performance reaches its peak varies across different categories. We believe that the peak point is determined by the quality of point clouds. For example, on the Car category, the tracked target may suffer from heavy occlusion and data missing, and thus it requires a large memory size to capture much shape information.

Table 6: **Ablation studies of different model components.** "De", "Q" and "PS" denote the decoupling design in DeFPM, coarse-to-fine score prediction and positive sampling strategy for non-rigid objects, respectively.

De	Q	PS	Car	Pedestrian	Van	Cyclist	Mean
	✓	✓	70.0/81.3	64.1/88.5	58.7/70.4	72.5/93.4	66.5/83.7
✓		✓	71.8/82.9	64.2/89.5	59.0/69.5	74.9/93.9	67.4/84.8
✓	✓		73.4/84.8	65.6/91.6	61.3/72.7	75.1/94.0	69.0/86.9
✓	✓	✓	**73.4/84.8**	**68.6/93.9**	**61.3/72.7**	**76.7/94.3**	**70.3/87.9**

Table 7: **Ablation studies of different localization heads.**

Loc	Car	Pedestrian	Van	Cyclist	Mean
RPN	67.2/81.1	53.5/85.5	52.0/62.4	61.3/90.2	59.8/81.5
PRM	69.0/81.4	59.0/88.4	54.0/64.5	71.6/92.6	63.4/83.2
V2B	72.6/84.2	61.1/87.9	55.6/64.7	71.2/93.8	66.1/84.3
X-RPN	70.4/81.9	64.9/91.3	55.1/64.6	72.1/93.2	64.7/84.7
BPLoc	**73.4/84.8**	**68.6/93.9**	**61.3/72.7**	**76.7/94.3**	**70.3/87.9**

Table 8: **Integration with Siamese-based network.**

Method	Car	Pedestrian	Van	Cyclist	Mean
CXTrack	69.1/81.6	67.0/91.5	60.0/71.8	**74.2/94.3**	67.5/85.3
CXTrack†	**72.8/84.5**	**67.7/92.1**	**61.3/72.7**	74.1/94.1	**69.6/87.0**
Improvement	↑3.7/↑2.9	↑0.7/↑0.6	↑1.3/↑0.9	↓0.1/↓0.2	↑2.1/↑1.6

†: integrated with BPLocNet

Model components. Tab. 6 presents ablation studies of MBPTrack on KITTI to gain a better understanding of its model designs. We investigate the impact of the decoupling design in DeFPM, the coarse-to-fine score prediction and the positve sampling training strategy via separate ablation experiments. Notably, we add targetness mask embedding to $X^{(l-1)}$ and $X_m^{(l-1)}$ before cross-attention to ablate the decoupling design, in which DeFPM is degraded to a one-branch transformer. Although the effectiveness of different components varies across categories, removing any of them leads to an obvious decline in terms of average metrics.

Localization head. We compare our proposed BPLocNet and other commonly-adopted localization heads on KITTI, including point-based (RPN [22], PRM [36], X-RPN [31]) and voxel-based (V2B [10]) methods. The results are shown in Tab. 7. BPLocNet consistently outperforms the alternative designs on all categories. We further integrate BPLocNet with a previous Siamese-based method CXTrack [31] to explore its generalization ability. Tab. 8 shows obvious performance gain by using BPLocNet, especially on the Car category (72.8/84.5 v.s. 69.1/81.6). For cars that have simple shapes and suffer from self-occlusions, box-prior sampling provides a strong shape prior to the localization task, thereby leading to better performance than the point-based X-RPN adopted in CXTrack [31]

5. Conclusion

We propose a memory-based tracker, named MBPTrack, to address the appearance variation and size difference problems in 3D single object tracking. MBPTrack employs a decoupling feature propagation module to exploit rich information lying in historical frames, which is overlooked by previous Siamese-based methods. We also design a novel localization network, named BPLocNet, that leverages box priors to more accurately localize the tracked targets of different sizes. Extensive experiments on three large-scale datasets show our method surpasses previous state-of-the-art on tracked targets of varying sizes while maintaining high efficiency. The major limitation of our work is the inaccurate orientation prediction caused by inaccurate past predictions (Fig. 5, bottom right). Besides, our method achieves limited performance when the point cloud is extremely sparse. In the future, we would like to explicitly model the target motion to address these issues.

Acknowledgment This work was supported by the Natural Science Foundation of China (Project Number 61832016) and Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[2] Widodo Budiharto, Edy Irwansyah, Jarot Sembodo Suroso, and Alexander Agung Santoso Gunawan. Design of object tracking for military robot using PID controller and computer vision. *ICIC Express Letters*, 14(3):289–294, 2020.

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[4] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[5] Yubo Cui, Zheng Fang, Jiayao Shan, Zuoxu Gu, and Sifan Zhou. 3D object tracking with transformer. *arXiv preprint arXiv:2110.14921*, 2021.

[6] Zheng Fang, Sifan Zhou, Yubo Cui, and Sebastian Scherer. 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.

[7] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[8] Silvio Giancola, Jesus Zarzar, and Bernard Ghanem. Leveraging shape completion for 3D Siamese tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1359–1368, 2019.

[9] Zhiyang Guo, Yunyao Mao, Wengang Zhou, Min Wang, and Houqiang Li. CMT: Context-matching-guided transformer for 3D tracking in point clouds. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII*, pages 95–111. Springer, 2022.

[10] Le Hui, Lingpeng Wang, Mingmei Cheng, Jin Xie, and Jian Yang. 3D Siamese voxel-to-BEV tracker for sparse point clouds. *Advances in Neural Information Processing Systems*, 34:28714–28727, 2021.

[11] Le Hui, Lingpeng Wang, Linghua Tang, Kaihao Lan, Jin Xie, and Jian Yang. 3D Siamese transformer network for single object tracking on point clouds. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II*, pages 293–310. Springer, 2022.

[12] Muxi Jiang, Rui Li, Qisheng Liu, Yingjing Shi, and Esteban Tlelo-Cuautle. High speed long-term visual object tracking algorithm for real robot systems. *Neurocomputing*, 434:268–284, 2021.

[13] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pages 5156–5165. PMLR, 2020.

[14] Matej Kristan, Jiri Matas, Aleš Leonardis, Tomáš Vojíř, Roman Pflugfelder, Gustavo Fernandez, Georg Nebehay, Fatih Porikli, and Luka Čehovin. A novel performance evaluation methodology for single-target trackers. *IEEE transactions on pattern analysis and machine intelligence*, 38(11):2137–2155, 2016.

[15] H Kuang Chiu, A Prioletti, J Li, and J Bohg. Probabilistic 3D multi-object tracking for autonomous driving. *ArXiv, vol. abs/2001.05673*, 2020.

[16] Kaihao Lan, Haobo Jiang, and Jin Xie. Temporal-aware Siamese tracker: Integrate temporal context for 3D object tracking. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 399–414, December 2022.

[17] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A frame-

work for attention-based permutation-invariant neural networks. In *International conference on machine learning*, pages 3744–3753. PMLR, 2019.

[18] Matthias Luber, Luciano Spinello, and Kai O Arras. People tracking in RGB-D data with on-line boosted target models. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3844–3849. IEEE, 2011.

[19] Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3D object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2906–2917, 2021.

[20] Ziqi Pang, Zhichao Li, and Naiyan Wang. Model-free vehicle tracking and state estimation in point cloud sequences. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8075–8082, 2021.

[21] Alessandro Pieropan, Niklas Bergström, Masatoshi Ishikawa, and Hedvig Kjellström. Robust 3D tracking of unknown objects. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2410–2417. IEEE, 2015.

[22] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep Hough voting for 3D object detection in point clouds. In *proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9277–9286, 2019.

[23] Haozhe Qi, Chen Feng, Zhiguo Cao, Feng Zhao, and Yang Xiao. P2B: Point-to-box network for 3D object tracking in point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6329–6338, 2020.

[24] Jiayao Shan, Sifan Zhou, Zheng Fang, and Yubo Cui. PTT: Point-track-transformer module for 3D single object tracking in point clouds. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1310–1316. IEEE, 2021.

[25] Luciano Spinello, Kai Arras, Rudolph Triebel, and Roland Siegwart. A layered approach to people detection in 3D range data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, pages 1625–1630, 2010.

[26] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020.

[27] Simen Thys, Wiebe Van Ranst, and Toon Goedemé. Fooling automated surveillance cameras: adversarial patches to attack person detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[29] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph CNN for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 38(5):1–12, 2019.

[30] Zhoutao Wang, Qian Xie, Yu-Kun Lai, Jing Wu, Kun Long, and Jun Wang. MLVSNet: Multi-level voting Siamese network for 3D visual tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3101–3110, 2021.

[31] Tian-Xing Xu, Yuan-Chen Guo, Yu-Kun Lai, and Song-Hai Zhang. CXTrack: Improving 3D point cloud tracking with contextual information. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1084–1093, June 2023.

[32] Zongxin Yang and Yi Yang. Decoupling features in hierarchical propagation for video object segmentation. *arXiv preprint arXiv:2210.09782*, 2022.

[33] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3D object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11784–11793, 2021.

[34] Chaoda Zheng, Xu Yan, Jiantao Gao, Weibing Zhao, Wei Zhang, Zhen Li, and Shuguang Cui. Box-aware feature enhancement for single object tracking on point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13199–13208, 2021.

[35] Chaoda Zheng, Xu Yan, Haiming Zhang, Baoyuan Wang, Shenghui Cheng, Shuguang Cui, and Zhen Li. Beyond 3D Siamese tracking: A motion-centric paradigm for 3D single object tracking in point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8111–8120, 2022.

[36] Changqing Zhou, Zhipeng Luo, Yueru Luo, Tianrui Liu, Liang Pan, Zhongang Cai, Haiyu Zhao, and Shijian Lu. PTTR: Relational 3D point cloud object tracking with transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8531–8540, 2022.