Temporal-aware Siamese Tracker: Integrate Temporal Context for 3D Object Tracking

Kaihao Lan, Haobo Jiang, and Jin Xie(✉)

Nanjing University of Science and Technology, Nanjing, China
{lkh, jiang.hao.bo, csjxie}@njust.edu.cn

Abstract. Learning discriminative target-specific feature representation for object localization is the core of the 3D Siamese object tracking algorithms. Current Siamese trackers focus on aggregating the target information from the latest template into the search area for target-specific feature construction, which presents the limited performance in the case of object occlusion or object missing. To this end, in this paper, we propose a novel temporal-aware Siamese tracking framework, where the rich target clue lying in a set of historical templates is integrated into the search area for reliable target-specific feature aggregation. Specifically, our method consists of three modules, including a template set sampling module, a temporal feature enhancement module and a temporal-aware feature aggregation module. In the template set sampling module, an effective scoring network is proposed to evaluate the tracking quality of the template so that the high-quality templates are collected to form the historical template set. Then, with the initial feature embeddings of the historical templates, the temporal feature enhancement module concatenates all template embeddings as a whole and then feeds them into a linear attention module for cross-template feature enhancement. Furthermore, the temporal-aware feature aggregation module aggregates the target clue lying in each template into the search area to construct multiple historical target-specific search-area features. Particularly, we follow the collection orders of the templates to fuse all generated target-specific features via an RNN-based module so that the fusion weight of the previous template information can be discounted to better fit the current tracking state. Finally, we feed the temporal fused target-specific feature into a modified CenterPoint detection head for target position regression. Extensive experiments on KITTI, NuScenes and waymo open datasets show the effectiveness of our proposed method. Source code is available at `https://github.com/tqsdyy/TAT`.

1 Introduction

Visual object tracking is a fundamental task in the computer vision field, and has achieved extensive applications such as autonomous driving [19] and robotics vision [4]. With the development of cheap LiDAR sensors, the point cloud-based 3D object tracking has obtained much more attention. Compared with the visual tracking using 2D images, point cloud data can effectively handle the challenges,

such as the changes in the light condition and the object size. Generally, with the high-quality 3D bounding box in the first frame, 3D single object tracking aims to continuously evaluate the state of the object throughout the tracking video sequence. However, the sparsity of the point cloud and the noise interference still hinder its applications in the real world.

In recent years, inspired by the successful applications of Siamese network [1] in 2D object tracking, 3D tracker focuses on exploiting the Siamese network paradigm for object tracking. As a pioneer, SC3D [8] is proposed to perform the 3D Siamese object tracking by matching the embedded shape information in the template to a large number of candidate proposals in the search area. However, SC3D is time-consuming and not end-to-end. To this end, Qi *et al.* [25] proposed the Point-to-Box network (P2B) for object tracking. A PointNet++ [24] is employed to extract the features of the template and search area, which are then utilized to construct the target-specific feature for object position regression via the VoteNet [23]. Based on P2B, Zheng *et al.* [37] further proposed a box-aware feature embedding to capture the prior shape information of the object for robust object location. In addition, Hui *et al.* [10] proposed a voxel-to-BEV Siamese tracker to improve the tracking performance in the cases of sparse point clouds. In summary, current Siamese trackers mainly focus on exploiting a single template for target-specific feature generation while ignoring the rich temporal context information lying in the set of the historical templates.

In this paper, we propose a simple yet powerful temporal-aware Siamese tracking framework, where the high-quality historical templates are collected to learn the discriminative target-specific feature for object localization. Our key idea is to utilize a powerful linear attention mechanism for temporal context learning among the historical templates, which is then integrated into the search area for robust temporal-aware target information aggregation and object localization.

Specifically, our framework consists of three modules, including a template set sampling module, a temporal feature enhancement module and a temporal-aware feature aggregation module. In the template set sampling module, with the template as input, a lightweight scoring network is designed to evaluate the 3DIoU between the template and the ground-truth target so that the high-quality template set can be obtained by sampling the templates with high 3DIoUs. Then, taking as input the initial features of historical templates, the temporal feature enhancement concatenates the point features of all templates as the whole and feeds them into a linear attention module for efficient cross-template feature enhancement. Furthermore, the temporal-aware feature aggregation module constructs a feature matching matrix between each template feature and the search-area feature, which guides the target information transferring in each template into the search area for the target-specific search-area feature generation. In particular, a RNN-based (Recurrent Neural Network) module is employed to fuse multiple target-specific features in the collection order of the templates, based on the intuition that the target information in the latter templates tend to own a higher correlation with the current tracking state. Finally, with the learned

target-specific feature, we utilize a modified CenterPoint [36] detection head for object position regression. Notably, benefitting from the target information aggregation from the historical templates, our method can still obtain effective target-specific feature representation in the case of occlusion or object missing. Extensive experiments verify the effectiveness of our proposed method.

The main contributions of our work are as follows:

- We propose a novel temporal-aware Siamese tracking framework, where the target information lying in the multiple historical templates is aggregated for the discriminative target-specific feature learning and the target localization.
- An effective 3DIoU-aware template selector is designed to collect high-quality templates as the historical template set.
- Our method can achieve state-of-the-art performance on multiple benchmarks and be robust in the cases of object occlusion or object missing.

2 Related Work

2.1 2D Object Tracking

In recent years, with the successful application of Siamese network [1] in the 2D object tracking, Siamese-based trackers [9,30,33,39] have become mainstream. The core idea of the Siamese tracker is to extract features from the template image and the current image by using the shared weight backbone network to ensure that the two images are mapped to the same feature space, and use the idea of similarity matching to locate the most similar part of the image to the template. However, the lack of depth information in RGB images makes it difficult for trackers to accurately estimate the depth of objects in the images. Therefore, scholars try to study the object tracking method based on RGB-D data, but it is still some efforts have been made on RGB-D object tracking. The subject approach used in RGB-D data-based trackers [2,22] is not much different from the 2D object tracking method, except that additional depth information enhances the tracker's ability to perceive depth information. Therefore, these methods still rely heavily on RGB information and still suffer from problems such as sensitivity to illumination changes and object size variations.

2.2 3D Single Object Tracking

Due to the characteristics of point cloud data, object tracking based on 3D point cloud can effectively avoid a series of problems in 2D image tracking field. Therefore, in recent years, much more work [6,27,32] focuses on 3D object tracking based on point clouds. As a pioneer, Giancola *et al.* [8] proposed a Siamese tracker named SC3D dedicated to 3D single object tracking (SOT). SC3D uses shape completion on the template to obtain the shape information of the target, generates a large number of candidate proposals in the search area and compares them with the template, and takes the most similar proposal as the current tracking result. Qi *et al.* [25] proposed a point-to-box network (P2B) for the problem that

SC3D cannot be trained end-to-end, P2B uses feature augmentation to enhance the perception ability of the search area to a specific template, and then uses VoteNet [23] to locate the target in the search area. Zheng *et al.* [37] proposed a box-aware module based on P2B to enhance the network's mining of bounding box prior information. Furthermore, to enhance the tracking performance of the tracker for sparse point clouds, V2B proposed by Hui *et al.* [10] uses a shape-aware feature learning module and uses a voxel-to-BEV detector to regress the object center. Lately, Jiang *et al.* [12] proposed a two-stage Siamese tracker named RDT, which uses point cloud registration to achieve robust feature matching between the template and potential targets in the search area.

In the 3D SOT task, existing Siamese trackers use a single-template matching mechanism for object localization, which ignores the rich temporal templates information of historical tracking results. This makes the trackers suffer from low-quality template feature representation in the presence of noise interference or occlusion, leading to the tracking failure. Therefore, we consider designing a novel Temporal-aware Siamese Tracker to associate temporal templates and sufficiently mine temporal context to improve the tracking robustness.

2.3 3D Multi Object Tracking.

Unlike SOT, most of the multi object tracking (MOT) algorithms follow the paradigm of "tracking-by-detection" [15,26,35], it including two stages: targets detection and targets association. Specifically, first, they use a detector to detect a large number of instance objects in each frame, and then use methods such as motion information to associate objects across frames. The difference between 3D-MOT and 2D-MOT is that 3D detection [17,28,23] and association [14] algorithms are used instead of 2D methods. As a pioneer, Weng *et al.* [34] uses PointRCNN [28] to detect instance targets per frame, and uses a 3D Kalman filter to predict object motion trajectories, finally using the Hungarian algorithm for detected objects matching. Recently, Luo *et al.* [18] attempted to unify detection and association into a unified framework, and achieved good results. In summary, 3D MOT pays more attention to the completeness of detection and the accuracy of association, and 3D SOT pays more attention to learning discriminative target-specific feature representation for object localization.

3 Approch

3.1 Problem Setting

In 3D single object tracking task, given the initial bounding box (BBox) of the object in the first frame, the tracker needs to continuously predict the BBoxes of the object throughout the tracking sequence. Specifically, an object BBox consists of nine parameters, including the object center coordinate (x, y, z), object size (l, w, h), and rotation angle (α, β, θ) (corresponding to three coordinate axes, respectively). Generally, in the 3D SOT field, we assume that the target

size is fixed and the rotation direction is just around the z-axis. Therefore, the estimation of the object states will only contain the center coordinates (x, y, z) and rotation angle θ.

Our temporal-aware Siamese tracking framework mainly consists of three modules, including the template set sampling module (Sec. 3.2), the temporal feature enhancement module (Sec. 3.3) and the temporal-aware feature aggregation module (Sec. 3.4).

3.2 Template Set Sampling

Following the Siamese tracking paradigm, the traditional Siamese tracker takes as input a single template point cloud $\boldsymbol{P}^t = \{\boldsymbol{p}_i^t \in \mathbb{R}^3 | i = 1, 2, ..., N\}$ and a search area point cloud $\boldsymbol{P}^s = \{\boldsymbol{p}_j^s \in \mathbb{R}^3 | j = 1, 2, ..., M\}$, and matches the closest target object to the template in the search area. Instead, we focus on extracting the rich temporal context information from a set of collected templates $\boldsymbol{T} = \{\boldsymbol{P}_1^t, \boldsymbol{P}_2^t, ..., \boldsymbol{P}_k^t\}$ for robust object localization. We consider three sampling mechanisms for template set generation, including the random sampling, the closest sampling and the template score ranking sampling, and in Sec. 4.3, we will discuss the effects of different ways of generating template set:

Random sampling. We randomly select k templates from the whole historical template buffer as the template set.

Closest sampling. We select k templates closest to the current frame as the template set, which tends to be more related to the current frame.

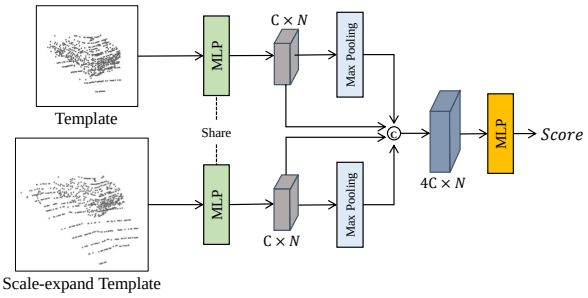

Fig. 1. The framework of the 3DIoU-aware template selection network.

Template score ranking sampling. To collect high-quality templates, we construct a scoring network to predict the 3DIoU score between each historical template and the ground-truth target, and then select k templates with the highest scores as the template set. Specifically, as shown in Fig. 1, we exploit the original template $\boldsymbol{P}^t$ and the scale-expanded template $\boldsymbol{P}^{tg} = \{\boldsymbol{p}_i^{tg} \in \mathbb{R}^3 | i = 1, 2, ..., N\}$ as the network input, where the scale-expanded template is used to provide the necessary context information for reliable 3DIoU prediction. Then, a weight-shared MLP is employed to extract their local point features $\boldsymbol{F}_{local}^t \in$

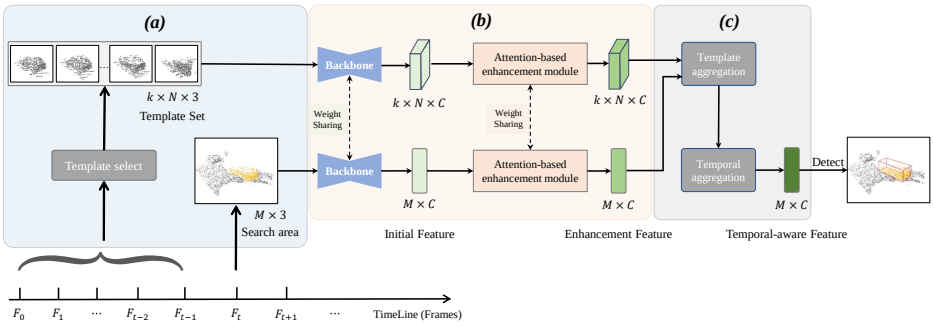

Fig. 2. The framework of our Temporal-aware Siamese Tracker. (a) **Template set sampling**: We first exploit the template selector to collect the high-quality template set. (b) **Siamese feature extraction**: Then, the weight-shared backbone network followed by a self-attention module is employed as the Siamese network to extract the point-level features of the template set and the search area. (c) **Target-specific feature fusion**: We integrate the target clue from k templates into the search area to learn k target-specific search area features, and an RNN-based fusion module is then employed to obtain the adaptively-fused target-specific feature. Finally, We pass the fused target-specific feature into a modified CenterPoint detector for object localization.

$\mathbb{R}^{C\times N}$ and $\boldsymbol{F}^{tg}_{local} \in \mathbb{R}^{C\times N}$, and meanwhile a max-pooling function is performed to extract their global feature $\boldsymbol{F}^{t}_{global} \in \mathbb{R}^{C\times N}$ and $\boldsymbol{F}^{tg}_{global} \in \mathbb{R}^{C\times N}$, respectively. Finally, taking as input the concatenated features of the local and global features, an MLP is used to predict the score of the template:

$$Score = \mathrm{MLP}(\mathrm{Cat}(\boldsymbol{F}^{t}_{local}, \boldsymbol{F}^{t}_{global}, \boldsymbol{F}^{tg}_{local}, \boldsymbol{F}^{tg}_{global})). \tag{1}$$

We use PointNet++ [24] as the backbone network to extract the initial features of the template set and the search area. We denote the obtained initial search-area feature as $\boldsymbol{S} \in \mathbb{R}^{M\times C}$, and the initial feature of l-th template in the template set as $\boldsymbol{T}^{l} \in \mathbb{R}^{N\times C}$, where C indicates the feature dimension.

3.3 Temporal Feature Enhancement

Temporal feature enhancement module aims to enhance the initial template features $\boldsymbol{T}^1, \boldsymbol{T}^2, ..., \boldsymbol{T}^k$ with the cross-template message passing based on the linear attention. In this section, we briefly introduce the linear attention mechanism, and then demonstrate how to exploit the temporal context information to enhance the feature representation of each template.

Linear-attention mechanism. The basic attention mechanism uses the dot-product attention [31] between the query $\boldsymbol{Q} \in \mathbb{R}^{N_q\times C}$ and key $\boldsymbol{K} \in \mathbb{R}^{N_k\times C}$ as the cross-attention weights for message passing. However, for the large-scale tasks, the dot-product attention is inefficient and usually needs high computational complexity for long-range relationship modeling. To relieve it, Katharopoulos *et al.* [13] proposed the linear attention that just needs a linear dot-product of

kernel feature maps for efficient attention weight generation. In detail, the linear attention can be defined as:

$$\mathrm{LA}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \phi(\boldsymbol{Q}) \left(\phi(\boldsymbol{K})^{\top}(\boldsymbol{V})\right) \tag{2}$$

where $\phi(\cdot) = \mathrm{elu}(\cdot) + 1$. The linear attention can also be extended to multi-head attention (denoted as "MultiHead-LA($\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$)") to capture richer feature representations. In the following, we employ such linear attention to integrate the temporal context information among the template set for the feature enhancement of each template.

Temporal feature enhancement. Given a set of initial template features, we first concatenate them to form an ensemble of the template features $\boldsymbol{T} = \mathrm{Concat}(\boldsymbol{T}^1, \boldsymbol{T}^2, ..., \boldsymbol{T}^k)$, where k is the template number of template set. Then, we treat them as a whole by reshaping $\boldsymbol{T} \in \mathbb{R}^{k \times N \times C}$ to $\boldsymbol{T} \in \mathbb{R}^{N_t \times C}$ ($N_t = k \times N$) to satisfy the input shape of linear-attention module. The feature enhancement can be formulated as below:

$$\boldsymbol{T}^{'} = \text{MultiHead-LA}(\boldsymbol{T} + \boldsymbol{T}_p, \boldsymbol{T} + \boldsymbol{T}_p, \boldsymbol{T} + \boldsymbol{T}_p) \tag{3}$$

where $\boldsymbol{T}^{'} \in \mathbb{R}^{N_t \times C}$ is the enhanced template feature with the linear-attention module, and $\boldsymbol{T}_p \in \mathbb{R}^{N_t \times C}$ is the coordinate embedding of the template points via a MLP. Furthermore, the enhanced feature $\boldsymbol{T}^{'}$ is added as the residual item to the initial feature $\boldsymbol{T}$, followed by an instance normalization operation $\mathrm{Ins.Norm}(\cdot)$:

$$\boldsymbol{T}^{''} = \mathrm{Ins.Norm}(\boldsymbol{T}^{'} + \boldsymbol{T}). \tag{4}$$

In addition, in order to improve the generalization ability of the network, we enhance $\boldsymbol{T}^{''}$ using a feed-forward neural network ($\mathrm{FFN}(\cdot)$) followed by a instance normalization:

$$\widehat{\boldsymbol{T}} = \mathrm{Ins.Norm}(\mathrm{FFN}(\boldsymbol{T}^{''}) + \boldsymbol{T}^{''}). \tag{5}$$

Finally, for sufficient message passing among the template set, we iteratively perform the feature enhancement above in m times to achieve the deeper temporal context information aggregation. We will discuss the performance changes of different attention-iteration times m and different attention heads n in Sec. 4.3. Also, to ensure the feature-space consistency between the template set and the search area, we share the linear-attention module to the search area feature for feature transformation, i.e. $\boldsymbol{S} \rightarrow \widehat{\boldsymbol{S}} \in \mathbb{R}^{M \times C}$.

3.4 Temporal-aware Feature Aggregation

The Siamese tracker focuses on constructing the feature similarity between the template and the search area, which guides the transferring of the target information from the template to the search area for the target-specific feature learning and the object localization. However, the current 3D Siamese trackers just exploit a single template for tracking while ignoring the rich template clue lying in the historical templates, resulting in low tracking performance in some challenging cases (e.g., the object occlusion and the missing).

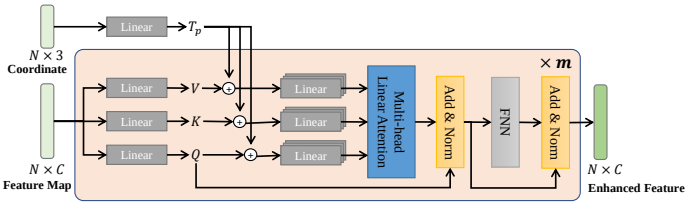

Fig. 3. Framework of the linear attention-based template enhancement module, which is iteratively performed m-times to achieve a richer temporal feature fusion.

Target-specific feature learning. Instead of using a single template, we focus on exploiting the sampled high-quality template set (in Sec. 3.2) for target-specific feature learning. Specifically, taking as input the enhanced template-set features $\widehat{\boldsymbol{T}} \in \mathbb{R}^{k\times N\times C}$ and search area feature $\widehat{\boldsymbol{S}} \in \mathbb{R}^{M\times C}$, we construct the feature similarity map via the Cosine similarity for each template $\widehat{\boldsymbol{T}}^l$ $(1 \le l \le k)$ and the search area $\widehat{\boldsymbol{S}}$:

$$\mathrm{Sim}^l_{i,j} = \frac{{\boldsymbol{t}_i}^T \cdot \boldsymbol{s}_j}{\|\boldsymbol{t}_i\|_2 \cdot \|\boldsymbol{s}_j\|_2}, \forall \boldsymbol{t}_i \in \widehat{\boldsymbol{T}}^l, \boldsymbol{s}_j \in \widehat{\boldsymbol{S}}. \tag{6}$$

Then, based on the feature similarity above, we use it to guide the target-specific search-area feature learning. Specifically, for j-th search-area point $\boldsymbol{p}^s_j$, we gather the target information in its most related template point $\boldsymbol{p}^t_{x^*}$ (index $x^* = \mathrm{argmax}_i\, \mathrm{Sim}^l_{i,j}$) of the l-th template, and use it to guide the transferring of the information into the $\boldsymbol{p}^s_j$. The target information of template point $\boldsymbol{p}^t_{x^*}$ consists of the point coordinate $\boldsymbol{p}^t_{x^*}$, enhanced point feature $\boldsymbol{t}^l_{x^*} \in \widehat{\boldsymbol{T}}^l$ and the similarity score $\mathrm{Sim}^l_{x^*,j}$. These target information will be sent to a MLP together with the corresponding search area feature $\boldsymbol{s}_j \in \widehat{\boldsymbol{S}}$ to build the fusion feature $\boldsymbol{s}^l_j = \mathrm{MLP}([\boldsymbol{s}_j, \boldsymbol{p}^t_{x^*}, \boldsymbol{t}^l_{x^*}, \mathrm{Sim}^l_{x^*,j}])$ of the j-th search-area point by the l-th template. Then, we can form the multiple target-specific features $\boldsymbol{S}^1_{fused}, \boldsymbol{S}^2_{fused}, \ldots, \boldsymbol{S}^k_{fused}$.

RNN-based temporal-aware feature fusion. With the generated multiple target-specific features above, we exploit a GRU network (Gated Recurrent Unit, a popular RNN network) to fuse them, where the GRU network can adaptively assign higher fusion weights on the target-specific features from the latter templates while discounting the fusion weights of the features from the previous templates. It is mainly based on the intuition that compared to the previous template information, the latter ones tend to own a higher correlation with the current tracking state, thereby can benefit the current tracking performance. Among them, for the k features $\boldsymbol{s}^1_j, \boldsymbol{s}^2_j, \ldots, \boldsymbol{s}^k_j$ at the j-th point in the search area, the process of GRU network to fuse the historical template information can be

formulated as:

$$\begin{aligned}
\boldsymbol{z}_j^t &= \sigma(\boldsymbol{W}^z \cdot [\boldsymbol{h}_j^{t-1}, \boldsymbol{s}_j^t]), \\
\boldsymbol{r}_j^t &= \sigma(\boldsymbol{W}^r \cdot [\boldsymbol{h}_j^{t-1}, \boldsymbol{s}_j^t]), \\
\widehat{\boldsymbol{h}}_j^t &= tanh(\boldsymbol{W} \cdot [\boldsymbol{r}_j^t * h_j^{t-1}, \boldsymbol{s}_j^t]), \\
\boldsymbol{h}_j^t &= (1 - \boldsymbol{z}_j^t) * \boldsymbol{h}_j^{t-1} + \boldsymbol{z}_j^t * \widehat{\boldsymbol{h}}_j^t,
\end{aligned} \tag{7}$$

where $\boldsymbol{W}, \boldsymbol{W}^z, \boldsymbol{W}^r$ are learnable parameter matrices, and $\boldsymbol{H} = \{\boldsymbol{h}_j^t \in \mathbb{R}^C | 1 \leq j \leq M, 1 \leq t \leq k\}$ is initialized to a zero matrix. We regard the output $\boldsymbol{H}^k \in \mathbb{R}^{M \times C}$ of the last layer as our final temporal-aware feature $\boldsymbol{S}_{final}$.

3.5 Loss Function

Templet scoring supervision. In the training phase, we first add the noise on the GT BBox of the target to generate the sample BBox and then generate the template by cropping the point cloud with the sample BBox. We use the IoU (Intersection over Union) between the GT BBox and the sample BBox as the GT score, and use the SmoothL1 Loss to supervise network training. Assuming that the GT score is S_{gt} and the predicted score of the network is S_{pred}, the loss of score supervision $\mathcal{L}_{score}$ can be written as:

$$\mathcal{L}_{score} = \text{SmoothL1}(S_{gt} - S_{pred}) \tag{8}$$

Detection head supervision. Based on the temporal-aware feature aggregation module, we obtain the temporal-aware fusion feature map, and we utilize the modified CenterPoint [36] detection network on this feature map to regress the target position. Following [10], we first voxelize the feature maps of each point by the averaging operation, and then use a stack of 3D convolutions to aggragte the features in the volumetric space. Next, we obtain the bird's eye view (BEV) feature map along the z-axis by max pooling operation. Finally, we aggregate the feature map using a stack of 2D convolution on the BEV feature map and use three different heads to regress the target position. Specifically, the three heads are 2D-center head, offset&rotation head, and z-axis head. We will use three losses to constrain them separately, and the details of the design can be found in [10], where we denote them as $\mathcal{L}_{detect}$.

The final loss function $\mathcal{L}$ is obtained by simply adding the two terms:

$$\mathcal{L} = \mathcal{L}_{score} + \mathcal{L}_{detect}. \tag{9}$$

4 Experiments

4.1 Experimental Settings

Implementation details. Following [25], we randomly sample $N = 512$ for template point cloud P^t and $M = 1024$ for search area point cloud, and sample $N = 512$ for scale-expand template point cloud in the template scoring module.

We set the size of the template set $k = 8$. In our attention enhancement module, we employ the number of iterations $n = 2$ and the number of heads of multi-heads attention $m = 4$. And for the temporal-aware feature aggregation module, we use the GRU network to associate k-temporal features, which can be defined and used via *torch.nn.GRUCell* in PyTorch [21]. We use the Adam [16] optimizer to update the network's parameters, and set the learning rate from initial 0.001 decayed by 0.2 every 6 epochs. The network will converge by training for about 30 epochs. We implement our model with PyTorch [21] and deploy all experiments on a server with TITAN RTX GPU and Intel i5 2.2GHz CPU.

Datasets. We use the KITTI [7], nuScenes [3] and waymo open [29] datasets for our experiments. Among them, the KITTI dataset has 21 video sequences. Following the P2B [25], we split the sequences into three parts: sequences 0-16 for training, 17-18 for validation, and 19-20 for testing. In addition, the nuScenes dataset with 700 training video sequences and 150 validation video sequences. Since the nuScenes dataset only labels the ground truth in key frames, following V2B [10], we use the official toolkit to interpolate the corresponding labels for the unlabeled frames. For the waymo open dataset (WOD), it is currently one of the largest outdoor point cloud datasets. Pang *et al.* [20] established a 3D SOT benchmark based on the WOD, which we will use directly.

Evaluation metrics. Following [25], we use *Success* and *Precision* criteria to measure the model performance. Specifically, *Success* is used to measure the IoU between the predicted BBox and the GT BBox, and *Precision* is used to measure the AUC (Area Under Curve) of the distance between the predicted BBox and the GT BBox centers from 0 to 2 meters.

Data pre-processing. The input to the core network consists of a template point cloud set, search area point cloud, and an additional scale-expand template point cloud is required for the template scoring module. For training, we enlarge the GT BBox of the previous frame by 2 meters and plus random offset and crop the search area from the current frame point cloud. In order to build the template set, we will randomly select k frame from the first frame of the current tracking sequence to the current frame, a small random noise is added to the GT BBox corresponding to each frame, and it is concatenated with the point cloud of the first frame as the template set. In addition, we will enlarge the BBox by 1 meter as the scale-expand template point cloud. For testing, we enlarge the predicted BBox of the previous frame by 2 meters in the current frame and collect the points inside to generate the search area. The template set of the current frame will evaluate all the tracking results of the historical frame through the template scoring module, and select the k frames with the highest scores to be spliced with the first frame respectively. The scale-expand template corresponding to each template will be obtained by expanding the respective BBox by 1 meter. For simplicity, in the subsequent discussion we name our **T**emporal-**A**ware Siamese **T**racker as **TAT**.

Table 1. The performance of different methods on the KITTI and nuScenes datasets. **Bold** and underline denote the best performance and the second-best performance of the compared methods, respectively. "Mean" denotes the average results of four categories.

	Metrics	*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 [8]	41.3	18.2	40.4	41.5	31.2	57.9	37.8	47.0	70.4	48.5
	P2B [25]	56.2	28.7	40.8	32.1	42.4	72.8	49.6	48.4	44.7	60.0
	LTTR [5]	65.0	33.2	35.8	66.2	48.7	77.1	56.8	45.6	89.9	65.8
	BAT [37]	60.5	42.1	52.4	33.7	51.2	77.7	70.1	67.0	45.4	72.8
	PTT [27]	67.8	44.9	43.6	37.2	55.1	81.8	72.0	52.5	47.3	74.2
	PTTR [38]	65.2	50.9	52.5	65.1	58.4	77.4	81.6	61.8	90.5	77.8
	V2B [10]	70.5	48.3	50.1	40.8	58.4	81.3	73.5	58.0	49.7	75.2
	STNet [11]	72.1	49.9	58.0	73.5	61.3	**84.0**	77.2	**70.6**	93.7	80.1
	TAT (ours)	**72.2**	**57.4**	**58.9**	**74.2**	**64.7**	83.3	**84.4**	69.2	**93.9**	**82.8**
	Category	Car	Pedestrian	Truck	Bicycle	Mean	Car	Pedestrian	Truck	Bicycle	Mean
	Frame Num.	15578	8019	3710	501	27808	15578	8019	3710	501	27808
nuScenes	SC3D [8]	23.9	13.6	28.9	16.1	21.5	26.2	15.1	26.4	18.5	22.9
	P2B [25]	32.7	18.1	28.1	18.5	27.6	35.5	25.0	25.8	**23.9**	30.9
	BAT [37]	32.9	19.6	29.4	17.8	28.3	35.3	**30.3**	26.2	21.9	32.4
	V2B [10]	36.5	19.4	30.8	18.9	30.5	39.0	26.9	28.6	22.0	33.8
	TAT (ours)	**36.8**	**20.7**	**32.2**	**19.0**	**31.2**	**39.6**	29.5	**28.9**	22.6	**34.9**

4.2 Result

Evaluation on KITTI dataset. Following [25], we select the four most representative categories from all object categories in the KITTI dataset [7] for our experiments, including Car, Pedestrian, Van and Cyclist. We compare our method with the previous state-of-art approaches [5,8,10,11,25,27,37,38]. Each of these methods have published their results on KITTI, and we use them directly. As shown at the top of Tab. 1, our method outperforms other methods in most metrics. For the mean result of the four categories, our method can improve the *Success* from 61.3(STNet) to 64.7(TAT) and *Precision* from 80.1(STNet) to 82.8(TAT), which boosted by 5.5% and 3.4%, respectively.

Evaluation on nuScenes dataset. We compare our method with the most typical four Siamese trackers [8,10,25,37] on the Car, Pedestrian, Truck and Bicycle categories of nuScenes dataset [3]. Following [10], since the nuScenes dataset is only labeled on key frames, we only report the performance evaluation on key frames. Compared with the KITTI dataset, the scenes of the nuScenes dataset are more complex and diverse, and the point cloud is more sparse, which greatly increases the challenge of the 3D SOT task. Nonetheless, as shown at the bottom of Tab. 1, our method can still achieve the best performance on the mean results of the four categories.

Evaluation on generalization ability. To evaluate the generalization ability of the model, we directly use the model trained on the corresponding classes of the KITTI dataset to evaluate its tracking performance on the WOD. Among them, the corresponding categories between WOD and KITTI are Vehi-

Table 2. The performance of different methods on the waymo open dataset. Each category is divided into three levels of difficulty: Easy, Medium and Hard. "Mean" denotes the average results of three difficulty.

	Category	Vehicle				Pedestrian			
	Split	Easy	Medium	Hard	Mean	Easy	Medium	Hard	Mean
	Frame Num.	67832	61252	56647	185731	85280	82253	74219	241752
Success	P2B [25]	57.1	52.0	47.9	52.6	18.1	17.8	17.7	17.9
	BAT [37]	61.0	53.3	48.9	54.7	19.3	17.8	17.2	18.2
	V2B [10]	64.5	55.1	52.0	57.6	27.9	22.5	20.1	23.7
	TAT (ours)	**66.0**	**56.6**	**52.9**	**58.9**	**32.1**	**25.6**	**21.8**	**26.7**
Precision	P2B [25]	65.4	60.7	58.5	61.7	30.8	30.0	29.3	30.1
	BAT [37]	68.3	60.9	57.8	62.7	32.6	29.8	28.3	30.3
	V2B [10]	71.5	63.2	62.0	65.9	43.9	36.2	33.1	37.9
	TAT (ours)	**72.6**	**64.2**	**62.5**	**66.7**	**49.5**	**40.3**	**35.9**	**42.2**

cle → Car, Pedestrian → Pedestrian respectively. As shown in the Tab. 2, our method still shows excellent performance, which verifies that our model still has an advantage in generalization ability.

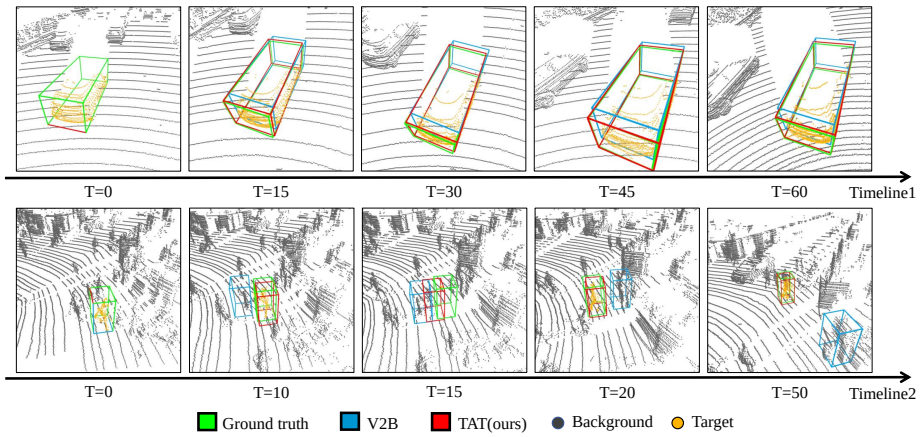

Fig. 4. The sequence tracking visualization of V2B and our TAT on the KITTI dataset. We color the GT BBoxes in green, while the BBoxes predicted by TAT and V2B are colored red and skyblue, respectively. In addition, we mark the points of target object in orange for better identification from the background.

Visualization. As shown in Fig. 4, we show the visualization results of sequence tracking of car and pedestrian on the KITTI dataset. For the car, our method can achieve more precise localization. For the pedestrian, the existing Siamese trackers (such as V2B) are easy to match incorrectly when multiple pedestrians are adjacent or occluded. However, our method can accurately localize the target object among multiple candidates in complex scenes. In the

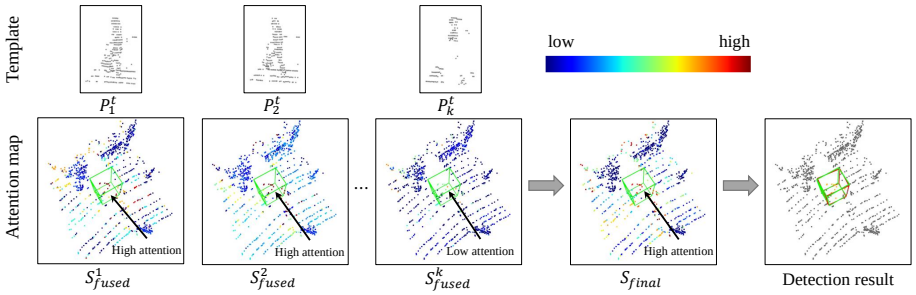

Fig. 5. The attention heatmap visualization of target-specific features, using historical templates to alleviate low-quality fusion features brought by recent low-quality templates, so that high-quality detection can still be achieved.

case of occlusion, since the point cloud of the real target surface is extremely sparse (T=15 in the bottom row), our method will also be biased to locate the surrounding wrong target. However, when the occlusion disappears (T=20 in the bottom row), our method can use the historical template context to quickly locate the correct target, which is impossible for ordinary Siamese trackers.

In addition, we provide an attention heatmap of fused features for this example to explain why our TAT can achieve high-quality tracking even after occlusion. As shown in the Fig. 5, although the recent template is of poor quality due to the previous occlusion, the historical temporal context allows us to obtain high matching confidence for the current frame.

More results. In addition, we provide more experimental results in the supplementary material, including quantitative results on the different sparse scenes, more categories visualizations of sequence tracking on the different datasets, running speed, and several visualizations of tracking videos. Please refer to the supplementary material for more experimental results and analyses.

4.3 Ablation Study

In this section, we design a rich ablation study to validate our proposed module and the effect of some hyperparameters on the results. We will simultaneously conduct experiments on the two main categories of car and pedestrian to provide comprehensive and reliable ablation study results.

Different collection strategies for template set. As we discussed earlier, the main purpose of introducing the template set is to use the historical successful tracking results to optimize the recent low-quality tracking. As shown at the bottom of Tab. 3, take the car category as an example, the performance of *Closest* sample method is significantly reduced by 2.6/3.2 (from 72.2/83.3 to 69.6/80.1), this further validates the effectiveness of our method for this situation. And compared with *Random* sample method, our proposed template score selector can bring performance gains of 0.7/0.6 (from 71.5/82.7 to 72.2/83.3) points.

Table 3. Ablation study on different template collection strategies and components. CSTS: Collection Strategies for Template Set. TFEM: Temporal Feature Enhancement Module. TFAM: Temporal-aware Feature Aggregation Module.

CSTS			Components		Car		Pedestrian	
Random	*Closest*	*Score*	TFEM	TFAM	*Success*	*Precision*	*Success*	*Precision*
		√			68.1 (-4.1)	79.3 (-4.0)	51.4 (-6.0)	78.1 (-6.3)
		√	√		69.0 (-3.2)	80.6 (-2.7)	52.1 (-5.3)	81.6 (-2.8)
		√		√	70.2 (-2.0)	81.5 (-1.8)	54.4 (-3.0)	83.0 (-1.4)
√			√	√	71.5 (-0.7)	82.7 (-0.6)	55.3 (-2.1)	83.3 (-1.1)
	√		√	√	69.6 (-2.6)	80.1 (-3.2)	52.7 (-4.7)	81.8 (-2.6)
		√	√	√	**72.2**	**83.3**	**57.4**	**84.4**

Temporal-aware components. The temporal-aware components consist of temporal feature enhancement module and temporal-aware feature aggregation module. For the temporal feature enhancement module, the dimensions of the input and output features are consistent, and we can directly remove it to verify the effectiveness of the module. For the temporal-aware feature aggregations module, we use the RNN-based network to associate the k fusion feature maps of different time series. In addition, we can also ignore the temporal relationship between templates, and use an MLP to obtain the final fusion feature map after concat k feature maps. As shown in the upper part of Tab. 3, take the car category as an example, these two modules can bring performance improvement of 2.0/1.8 (from 70.2/81.5 to 72.2/83.3) and 3.2/2.7 (from 69.0/80.6 to 72.2/83.3) respectively. These ablation experiments effectively verify that our proposed module can utilize the temporal context more effectively.

More ablation studies. We also investigate the effect of different numbers of templates and hyperparameters of attention modules on the results. Based on the experiments, the number of template sets is 8, the number of heads for multi-head attention is 4, and the number of iterations of the template enhancement module is 2 is a good experimental setting. For more experimental data and analysis, please refer to Supplementary Materials.

5 Conclusions

In this paper, we proposed a simple yet powerful Temporal-aware Siamese tracking framework, where we introduce the temporal feature enhancement module and the temporal-aware feature aggregation module into the architecture of the Siamese 3D tracking methods. Our method optimizes the current tracking process by correlating multi-frame template set and making full use of temporal context. In addition, we designed a simple yet effective 3DIoU-aware template selector to build a high-quality temporal template set. Our proposed method significantly improves the performance of 3D SOT on several benchmark datasets.

Acknowledgements This work was supported by the National Science Fund of China (Grant No. 61876084).

References

1. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.: Fully-convolutional siamese networks for object tracking. In: ECCV (2016)
2. Bibi, A., Zhang, T., Ghanem, B.: 3d part-based sparse tracker with automatic synchronization and registration. In: CVPR (2016)
3. 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. In: CVPR (2020)
4. Comport, A.I., Marchand, É., Chaumette, F.: Robust model-based tracking for robot vision. In: IROS (2004)
5. Cui, Y., Fang, Z., Shan, J., Gu, Z., Zhou, S.: 3d object tracking with transformer. arXiv preprint arXiv:2110.14921 (2021)
6. 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 (2020)
7. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: CVPR (2012)
8. Giancola, S., Zarzar, J., Ghanem, B.: Leveraging shape completion for 3d siamese tracking. In: CVPR (2019)
9. Guo, Q., Feng, W., Zhou, C., Huang, R., Wan, L., Wang, S.: Learning dynamic siamese network for visual object tracking. In: ICCV (2017)
10. Hui, L., Wang, L., Cheng, M., Xie, J., Yang, J.: 3d siamese voxel-to-bev tracker for sparse point clouds. In: NeurIPS (2021)
11. Hui, L., Wang, L., Tang, L., Lan, K., Xie, J., Yang, J.: 3d siamese transformer network for single object tracking on point clouds. In: ECCV (2022)
12. Jiang, H., Lan, K., Hui, L., Li, G., Xie, J., Yang, J.: Point cloud registration-driven robust feature matching for 3d siamese object tracking. arXiv preprint arXiv:2209.06395 (2022)
13. Katharopoulos, A., Vyas, A., Pappas, N., Fleuret, F.: Transformers are RNNs: Fast autoregressive transformers with linear attention. In: ICML (2020)
14. Kelly, A.: A 3d state space formulation of a navigation kalman filter for autonomous vehicles. Tech. rep., CARNEGIE-MELLON UNIV PITTSBURGH PA ROBOTICS INST (1994)
15. Kim, A., Ošep, A., Leal-Taixé, L.: Eagermot: 3d multi-object tracking via sensor fusion. In: ICRA (2021)
16. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)
17. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: Fast encoders for object detection from point clouds. In: CVPR (2019)
18. Luo, C., Yang, X., Yuille, A.: Exploring simple 3d multi-object tracking for autonomous driving. In: ICCV (2021)
19. Luo, W., Yang, B., Urtasun, R.: Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In: CVPR (2018)
20. Pang, Z., Li, Z., Wang, N.: Model-free vehicle tracking and state estimation in point cloud sequences. In: IROS (2021)
21. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. NeurIPS (2019)

22. Pieropan, A., Bergström, N., Ishikawa, M., Kjellström, H.: Robust 3d tracking of unknown objects. In: ICRA (2015)
23. Qi, C.R., Litany, O., He, K., Guibas, L.J.: Deep hough voting for 3d object detection in point clouds. In: ICCV (2019)
24. 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)
25. 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)
26. Scheidegger, S., Benjaminsson, J., Rosenberg, E., Krishnan, A., Granström, K.: Mono-camera 3d multi-object tracking using deep learning detections and pmbm filtering. In: IV (2018)
27. Shan, J., Zhou, S., Fang, Z., Cui, Y.: Ptt: Point-track-transformer module for 3d single object tracking in point clouds. In: IROS (2021)
28. Shi, S., Wang, X., Li, H.: Pointrcnn: 3d object proposal generation and detection from point cloud. In: CVPR (2019)
29. 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)
30. Tao, R., Gavves, E., Smeulders, A.W.: Siamese instance search for tracking. In: CVPR (2016)
31. 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)
32. Wang, L., Hui, L., Xie, J.: Facilitating 3d object tracking in point clouds with image semantics and geometry. In: PRCV (2021)
33. Wang, N., Zhou, W., Wang, J., Li, H.: Transformer meets tracker: Exploiting temporal context for robust visual tracking. In: CVPR (2021)
34. Weng, X., Kitani, K.: A baseline for 3d multi-object tracking. arXiv preprint arXiv:1907.03961 (2019)
35. Wu, H., Han, W., Wen, C., Li, X., Wang, C.: 3d multi-object tracking in point clouds based on prediction confidence-guided data association. IEEE Transactions on Intelligent Transportation Systems (2021)
36. Yin, T., Zhou, X., Krahenbuhl, P.: Center-based 3d object detection and tracking. In: CVPR (2021)
37. 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)
38. 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. In: CVPR (2022)
39. Zhu, Z., Wang, Q., Li, B., Wu, W., Yan, J., Hu, W.: Distractor-aware siamese networks for visual object tracking. In: ECCV (2018)