OST: Efficient One-Stream Network for 3D Single Object Tracking in Point Clouds

Xiantong Zhao, Yinan Han, Shengjing Tian, Jian Liu, and Xiuping Liu

Abstract—Although recent Siamese network-based trackers have achieved impressive perceptual accuracy for single object tracking in LiDAR point clouds, they usually utilized heavy correlation operations to capture category-level characteristics only, and overlook the inherent merit of arbitrariness in contrast to multiple object tracking. In this work, we propose a radically novel one-stream network with the strength of the instance-level encoding, which avoids the correlation operations occurring in previous Siamese network, thus considerably reducing the computational effort. In particular, the proposed method mainly consists of a Template-aware Transformer Module (TTM) and a Multi-scale Feature Aggregation (MFA) module capable of fusing spatial and semantic information. The TTM stitches the specified template and the search region together and leverages an attention mechanism to establish the information flow, breaking the previous pattern of independent extraction-and-correlation. As a result, this module makes it possible to directly generate template-aware features that are suitable for the arbitrary and continuously changing nature of the target, enabling the model to deal with unseen categories. In addition, the MFA is proposed to make spatial and semantic information complementary to each other, which is characterized by reverse directional feature propagation that aggregates information from shallow to deep layers. Extensive experiments on KITTI and nuScenes demonstrate that our method has achieved considerable performance not only for class-specific tracking but also for class-agnostic tracking with less computation and higher efficiency.

Index Terms—3D single object tracking, one-stream network, class-agnostic, template-aware, multi-scale.

I. INTRODUCTION

OBJECT tracking, as an important means of perceiving the external environment for the agent, has great significance in many fields, such as autonomous driving, robotics vision, and visual surveillance [1], [2], [3]. Recently, the development and popularity of light detection and ranging (LiDAR) bring new energy to the tracking research community, whose research subjects have expanded from classical visual tracking [4], [5], [6] to challenging 3D object tracking in point clouds. In general, 3D tracking tasks can be roughly divided into 3D single object tracking (3D SOT) and 3D multiple object tracking (3D MOT). The former is designed to locate the object arbitrarily specified in the first frame, while the latter concentrates on finding all class-specific objects through the detection-and-association mechanism. In the following discussion, the research objective will focus on solving some pain points existing in popular 3D single object tracking paradigms.

There are three mainstream schemes for solving 3D SOT problems in these communities. The first one deals with this task using purely appearance models, such as SC3D [7], which learns a general feature matching as a 2D counterpart of SiamFC [5]. Nevertheless, the model cannot conduct end-to-end train limited by multi-frames polymerization and feature matching. The second is the family of motion-based methods, like MM-track [8], SEDT [9], Register-driven [10], which utilized the aligned features or refine the coarse bounding box with target relative motion. Although the adoption of motion clues has greatly improved tracking accuracy, the correct prediction of target relative motion in the sparse scenes is also an arduous task. The last one based on feature fusion, relying on its lean structure and easiness to train, dominates the trend of this task. These works integrate the 3D detection head (VoteNet [11], VoxelNet [12]) hinged on a powerful feature embedding between the template and search region, typical examples include P2B [13], BAT [14], PTT [15], V2B [16] and so on. Even though leaving an impressive performance, all above frameworks still struggle with two dilemmas.

On the one hand, the Siamese network has two branches for template and search regions with shared weights. In the process of feature extraction, the branch of the search region can’t feel the template information. Siamese-based methods suffer from two inherent barriers, which are caused by this noninteractive feature extraction of the template and search region. Firstly, the features extracted by the Siamese backbone are only able to make category-level rather than instance-level distinctions due to the absence of the template information, resulting in insufficient discernment between targets and background distractors. For example, when several similar objects simultaneously appear in the search region, Siamese-based trackers may be misled by the interferes, as shown in Fig. 1. Secondly, Siamese-based...
We introduce a multi-scale feature extraction module to ensure both local geometrical information and global semantic information complementary to each other, which is characterized by reverse directional feature propagation that aggregates information from shallow to deep layers.

We conduct some representative experiments on KITTI [18] and nuScenes [19] datasets. The proposed method considerably reduces the computational cost and thus achieves high training and inference speed, without the loss of the tracking accuracy under the class-specific setting compared to the state-of-the-art methods. And more notably, our one-stream design for 3D tracking obtains better performance under the class-agnostic setting.

II. RELATED WORK

A. 3D Siamese Tracking

Shape Completion 3D (SC3D) network [7] is the pioneering work in this category, in which Siamese tracker encodes the targets and search regions into latent representations, then the target can be determined by comparing the similarity of latent representations among hand-crafted proposals. After SC3D, Point-to-Box (P2B) network [13] is another milestone. It uses a Siamese PointNet++ [20] to encode the template and search region, then merges the template information into the down-sampled search region by target-specific feature augmentation module, and eventually a detection head based on VoteNet [11] consumes such augmented search region to regress the most appropriate bounding box. P2B solves these problems that SC3D cannot perform end-to-end training and candidate generation. Subsequently, in light of its potential improvement room, researchers proposed some variants, such as BAT [14], which embeds information of the bounding box into search seeds during the feature fusion process, and MLVSNNet [21], which conducts multi-level voting to deal with sparsity in point clouds. Despite outstanding performance at that time, its results are not state-of-the-art compared with the current cutting-edge methods. The latest Siamese tracking methods replace the voting head with Bird’s-Eye-View (BEV) head [16], which incorporates information of the bounding box into search seeds during the feature fusion process, and eventually a detection head based on VoteNet [11] consumes such augmented search region to regress the most appropriate bounding box. V2B [16] transfers fusion features from point to BEV via voxelization and max-pooling, then BEV features will be sent into some 2D-Convolution layers to regress the target bounding box. LTTR [27] transfers the template and search region features into BEV maps separately, and then conducts feature fusion on BEV through the sheer force of Transformer. As discussed above, all mentioned approaches perform well under the class-specific setting but seldom unattended under category-agnostic settings [28]. To gap this challenge, we propose a one-stream framework for 3D single object tracking which can capture target-wise rather than category-wise attributes so that their accuracy far exceeds other baseline methods under category-agnostic settings.

B. Transformer in Point Cloud

Benefiting from Transformer’s powerful encoding capacity, the precision of 2D visual tasks have been achieved great
Fig. 2. Overview of the proposed OST. After concatenating the template and the search region, we send it into the one-stream network consisting of a local encoding module, a template-aware Transformer module, and a multi-scale feature aggregation module. In this way, we can generate template-aware features for the search region. Afterwards, following the feature augmentation by the segmentation prior, we voxelized the feature tensor for proposal generation. The overall process is at the top of this figure, and the module details are at the bottom.

C. 2D Visual Tracking

In 2D visual tracking, several studies [33] [34] [35] have noted that siamese-based networks suffer from low efficiency and have poor discrimination ability under certain challenging scenarios. Addressing challenging tracking attributes, especially object deformation and background clutter, Sheng et al. [33] proposed a distilled Siamese tracking framework to learn from small, fast and accurate trackers (students), which capture critical knowledge from large Siamese trackers (teachers) by a teacher-students knowledge distillation model. Han et al. [34] proposed a learnable module to better capture the semantic correlation information and produce more discriminative features. Dong et al. [35] provided an in-depth analysis of Siamese-based trackers and took sequence-specific samples from the first frame as decisive samples to deal with the problem. In addition, some 2D one-stream frameworks [36], [37] have been proposed to perform their unique character in 2D object tracking. Ye et al. [36] released some Siamese-based framework’s problems: the extracted features lack the awareness of the target and have limited target-background discriminability. To tackle these issues, Ye et al. introduced a one-stream tracking framework that can bridge template-search image pairs with bidirectional information flows. Single Branch Transformer (SBT) [37] suppresses non-target features and obtains instance-varying features by extensively matching the features of the two images through a one-stream backbone. To sum up, one-stream frameworks possess the ability to extract more discriminative features than ordinary Siamese frameworks. Inspired by these works, we migrate this attribute to 3D single object tracking with an elaborately designed template-aware Transformer and multi-scale feature aggregation for point clouds.

III. Method

In 3D single object tracking, a tracker aims to locate the target at each frame of point cloud sequence. In general, the target is specified in the first frame to guide the tracking procedure in the following frames. Mathematically, given the template \( P_t \) and
search region $P_s$, which are composed of $N_t$ and $N_s$ coordinates \((x, y, z)\) respectively, a tracker is supposed to accurately predict the state of the specified object in the search region $P_s$. We make use of a 3D bounding box to represent the state of the target, which is parameterized by center \((x, y, z)\), size \((l, w, h)\), and yaw angle $\theta$. Considering the outdoor scenes captured by LiDAR generally does not deform the physical size of the rigid target, the tracking problem can be formulated as the following,

\[
\{x, y, z, \theta\} = \Phi(P_t, P_s),
\]

where $\Phi$ represents the tracking model. Many existing algorithms have been proposed to toward this problem. The Siamese-network based method focuses on learning shape features and applying correlation operation between the template and search region. This needs a high time consumption despite achieving impressive performance. Besides, the current methods focus only on tracking those object categories seen during training. We deem that they lack the ability to learn discriminative features to capture target information that is arbitrarily specified in the first frame. To overcome this shortcoming, we propose a novel method with the philosophy of one-stream. The proposed method can not only reduce computing costs by integrating feature extraction with relation modeling, but also enhance the ability to learn discriminative features by building an information flow between the template and search region.

An overview of the model is shown in Fig. 2. It includes three core parts: template-aware Transformer module (TTM), multi-scale feature aggregation (MFA), as well as a specific loss. In the subsection, we first introduce our one-stream architecture which is composed of cascaded TTM, and then present the details of MFA, finally the task-specific loss is described.

### A. One-Stream Architecture

Most current methods are based on Pointnet++[13], [14], [38] or 3D sparse convolution [25] to extract point cloud features, and continue the practice of Siamese networks with shared parameters as 2D tracking did. The core to this framework must rely on a powerful feature fusion module to embed the template information into the search region, allowing the generation of features that discriminate strongly in the foreground and background. Such fused features will further facilitate the subsequent verification of the target by the detection head. Different from these conventional workflows, we introduce a one-stream architecture, the standpoint behind which lies in two aspects: One is that the correlation between the template and search region is discarded for high efficiency; the other one is the jointly feature learning for the $P_t$ and $P_s$ using the Transformer instead of PointNet++ and sparse convolution.

**Template-aware Transformer Module:** Our one-stream network is composed of template-aware Transformer encoder module. Specifically, we sample 512 points $P_t \in \mathbb{R}^{512 \times 3}$ in the template and 1024 points $P_s \in \mathbb{R}^{1024 \times 3}$ in the search region, and we feed the points in template $P_t$ and search region $P_s$ to a one-stream feature extraction backbone. First, since the raw coordinate of each point cannot describe its local structure, we thus adopt graph convolution neural network (GCN) [39] to encode spatial features in the corresponding neighbor. And the resulting features of the template and search region are represented as $F_t \in \mathbb{R}^{512 \times D}$, $F_s \in \mathbb{R}^{1024 \times D}$, respectively, in which $D$ indicates the feature dimension. Next, as the input of the one-stream network, we need to concatenate both coordinates and features of the template and search region, yielding $P_c = [P_t; P_s]$ and $F_c = [F_t; F_s]$. Then, the Transformer encoder $\phi$ takes the $P_c$ and $F_c$ as inputs to establish an interactive flow between $F_t$ and $F_s$. We can simply formulate this procedure as

\[
\hat{F}_c = \phi(P_c, F_c),
\]

where $\hat{F}_c$ is a template-aware feature produced by one Transformer block.

The inside mechanism of the Transformer encoder $\phi$ is illustrated in the Fig. 3(a). In our design, we iteratively implement this block with $\tau$ times, mathematically, it can be written as:

\[
\hat{F}_c^{(\tau)} = \phi^{(i)}(P_c^{(i-1)}, \hat{F}_c^{(i-1)}),
\]

where the superscript $i \in \{1, 2, \ldots, \tau\}$ represents the $i$-th Transformer block and $\hat{F}_c^{(0)} = F_c$. In our method, we perform it iteratively three times, i.e., $\tau = 3$. Note that the output in the previous iteration will replace the input in the next iteration. In the ablation study, we will fully explore the impact of the number of $\tau$.

**Analyze of Transformer module:** As we mentioned in the previous section, the Transformer in our method can establish information flow between the template and the search region. In fact, the self-attention mechanism in Transformer plays an important role. For simplification, we take the (2) as an example and further analyze the intrinsic reasons for it. The output of the self-attention mechanism in our method can be written as:

\[
\hat{F}_c = \phi(P_c, F_c)
\]

\[
= \text{Softmax} \left( \frac{Q_c K_c^T}{\sqrt{d}} \right) \cdot V_c,
\]

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
where \(d\) is the feature dimension of each attention head; \(Q_c, K_c, V_c\) are query, key and value matrices Transformed from \(F_c + P E(P_c)\) by the linear layers. Herein, \(PE(\cdot)\) is the a learnable position embedding.

Since the \(P_c\) and \(F_c\) derive from the concatenation of the template and search region, we can also decompose the \(Q_c, K_c, \) and \(V_c\) as the following formulation:

\[
Q_c = [Q_t; Q_s], \\
K_c = [K_t; K_s], \\
V_c = [V_t; V_s],
\]

(5)

where the subscripts \(t\) and \(s\) denote the template and search region separately.

Substituting the \(Q_c, K_c, \) and \(V_c\) with the above equation, we can further obtain

\[
\hat{F}_c = Softmax \left( \frac{[Q_t; Q_s][K_t; K_s]^T}{\sqrt{d_k}} \right) \cdot [V_t; V_s]
\]

\[
\triangleq [W_{tt}, W_{ts}; W_{st}, W_{ss}] \cdot [V_t; V_s]
\]

\[
= [W_{tt}V_t + W_{ts}V_s; W_{st}V_t + W_{ss}V_s],
\]

(6)

where \([W_{tt}, W_{ts}; W_{st}, W_{ss}]\) is the attention weight, and the \(W_{ab}, a, b \in \{t, s\}\) is a measure of similarity between \(a\) and \(b\).

It can be seen that this one-stream structure based on the Transformer has essentially fulfilled the extraction of the template (i.e., \(W_{tt}V_t\)) and the search region (i.e., \(W_{ss}V_s\)), and realized the information interaction between them (i.e., \(W_{ts}V_s \) and \(W_{st}V_t\)). Therefore, we can leverage this template-aware features to complete the verification of the target, without extra correlation module.

### B. Multi-Scale Feature Aggregation

In order to make spatial and semantic information complement each other for tracking task, we elaborately design a multi-scale feature aggregation module. The spatial shape features are beneficial for the network to precisely infer the bounding boxes, while semantic discrimination is able to inform the network of the presence of the target. However, with the deepening of the neural network, the semantic information of the extracted features gets stronger and the spatial receptive field is also larger. Considering that semantic discriminability is more crucial for the judgment of the foreground and background of the target in the search region, especially in the tracking task of unobserved objects, we aim to obtain more spatial information while ensuring the semantic discrimination of features. So, we designed a specific multi-scale feature aggregation method to supplement the semantic information with spatial information by inverse sampling.

For the input point cloud pairs (template and search region), their coordinates are \(P_t \in \mathbb{R}^{512\times 3}, P_s \in \mathbb{R}^{1024 \times 3}\), and their features are \(F_t, F_s\). After the template-aware Transformer module, we can calculate the feature set \([\hat{F}_c^{(1)}, \hat{F}_c^{(2)}, \hat{F}_c^{(3)}]\) from different layers of Transformer. According to (5), by truncating the \(\hat{F}_c^{(i)}\), we can obtain a template-aware feature \(\hat{F}_s^{(i)}\) for the search region in each Transformer block. Next, we down-sample search region points from each Transformer layer by farthest point sampling method. Different from the common method, we sample 256 points \(\hat{F}_s^{(1)}\) in the first TTM layer and 512 points \(\hat{F}_s^{(2)}\) in the second Transformer layer. Then we gather the feature \(\hat{F}_s^{(1)}, \hat{F}_s^{(2)}\) of sampled points from \(\hat{F}_c^{(1)}\) and \(\hat{F}_c^{(2)}\). As shown in Fig. 3, we aggregate different scales features of search region using Feature Propagation [20] and obtain the final feature \(\hat{F}_s^{(3)}\) from this module.

\[
\hat{F}_s^{(2)} = FP \left( \hat{F}_c^{(1)}, \hat{F}_s^{(2)} \right),
\]

\[
\hat{F}_s^{(3)} = FP \left( \hat{F}_c^{(2)}, \hat{F}_s^{(3)} \right).
\]

(7)

Multi-scale features in different layers capture the information at different scales of the point cloud. In our method, as the features flowing in the template-aware Transformer module, the semantic information is stronger and the spatial information weaker. Our MFA module embeds the spatial information of the front two layers into the final layer. Through multi-scale feature aggregation, spatial information can be effectively supplied to the feature.

### C. Loss Function

As shown at the bottom of Fig. 4, we take the Voxel-to-BEV method in [16] as our region proposal network. It first voxelizes the points in the search region with size of \([L, W, H]\). Remaining the original feature dimension, we can get the feature map with size of \([D, L, W, H]\). Then these voxels are projected onto the bird’s eye view (BEV), generating the feature map with size of \([D, W, H]\). In the end, proposal prediction is performed on the BEV feature map. More importantly, to improve the performance of this subnetwork, we introduce a new task-specific loss that considers the segmentation regularization to promote the verification of the target. The insight behind this design is that the points lying on the surface of the target is a critical clue which can provide the discriminativeness for the subsequent target location. Therefore, we enhance semantic features of the search region using segmentation instead of completion. In the Section IV-E, we will investigate this effect comprehensively.
The final prediction head includes the center loss $L_{\text{center}}$ which controls the regression of the center position on the BEV, the offset loss $L_{\text{offset}}$ which constrains the movement vector of the target, the Z-axis loss $L_{z\text{-axis}}$ which reflects the difference in height of the target, as well as the segmentation loss $L_{\text{seg}}$ which promotes the target verification by the discriminative score. The overall loss function can be written as:

$$L_{\text{total}} = \lambda_1 L_{\text{seg}} + \lambda_2 L_{\text{center}} + \lambda_3 L_{\text{offset}} + \lambda_4 L_{z\text{-axis}},$$

where $\lambda_1, \lambda_2, \lambda_3,$ and $\lambda_4$ are the hyper-parameter for segmentation, center regression, movement vector regression, and z-axis height regression, respectively. In our experiments, we empirically set these parameters as $\lambda_1 = 1.0, \lambda_2 = 1.0, \lambda_3 = 1.0,$ and $\lambda_4 = 2.0.$ In the following, we will describe each loss term in detail.

**Center regression on BEV:** Following V2B [16], the target is located by the peak of the heatmap $H \in R^{H \times W \times 1},$ we project the 3D ground-truth from the bird’s eye view, and can naturally generate the 2D target center and rectangle for the regression of detection head. For pixel $p_{ij}$ in BEV, (1), if the pixel is the 2D ground truth center, $H_{ij} = 1,$ (2) if the Pixel is not the 2D ground truth center but within the 2D ground truth bounding box, $H_{ij} = \frac{1}{d}$ where $d$ is the distance from this pixel to 2D ground truth center. (3) if the pixel is out of the 2D ground truth bounding box, $H_{ij} = 0.$ The loss can be calculated by modified focal loss [40], [41], [42] as follows:

$$L_{\text{center}} = - \sum \{ I[H_{ij} = 1] \cdot (1 - \hat{H}_{ij})^\alpha \log(\hat{H}_{ij}) + I[H_{ij} \neq 1] \cdot (1 - H_{ij})^\beta (\hat{H}_{ij})^\gamma \log(1 - \hat{H}_{ij}) \},$$

where $\hat{H}_{ij}$ is the score whether the pixel is the center point predicted by bev head. Where $I(\text{cond})$ is the indicator function. If $\text{cond}$ is true, then $I(\text{cond}) = 1,$ otherwise 0. Besides, we set $\alpha = 2$ and $\beta = 4$ in all experiments.

**Offset regression:** $L_{\text{center}}$ constrains the target center to a discrete center on BEV, which is inaccurate for predicting the target center. We consider regressing the target center from discrete to the continuous 2D ground truth center. We regress the offset from the pixel center position to the continuous position that the 3D object center mapped to the BEV. We select a square area with the radius $r$ around object center pixel in the offset regression map. And we train the offset using $L_1$ loss [43] meanwhile train the rotation.

$$L_{\text{offset}} = \sum_{\delta = -r}^{r} \sum_{\gamma = -r}^{r} \left| \hat{O}_{\gamma + (\delta + \gamma)} - [\hat{c} - \bar{p} + (\delta + \gamma), \theta] \right|,$$

where $\hat{O} \in R^{H \times W \times 3}$ is the offset and rotation predicted by BEV head. $\bar{c}$ and $c$ mean the discrete and continuous position of the ground truth center.

**z-axis regression:** BEV head regresses the z-axis position of the target center from the BEV feature map, and predict a map $\hat{Z} \in R^{H \times W \times 1}.$ We compute the error in the z-axis center using the $L_1$ loss:

$$L_{z\text{-axis}} = \left| \hat{Z} - Z \right|,$$

where $\bar{c}$ is the discrete object center and $z$ is the ground truth of the z-axis center.

**Segmentation:** As shown at the top of Fig. 4, we learn a MLP to predict a probability $S \in R^{1024}$ for each point in search region. $s_j \in S$ denotes the score whether a point $p_j \in R^3$ is a target point. Specifically, $S$ is constrained by a standard binary cross entropy loss. The points are considered positive if they are located in the ground truth bounding box, otherwise these points are negative. For enhancing the discriminative features, we concatenate the probability $s_j,$ coordinate $p_j$ and feature $\vec{F}_{s,j}(3)$ as \{ $s_j; p_j; \vec{F}_{s,j}$ \} $\in R^{1+3+D}$ for each point in search region. Then enhanced features are feed into Voxel-to-Bev head to predict proposals.

### IV. EXPERIMENTS

To verify the capabilities of our method, we organize the experiments into four parts: class-specific tracking, class-agnostic tracking, computational cost, and an ablation study. In this section, we first introduce our implementation details. Then we report the results of our method on different benchmark datasets with comparisons several state-of-the-art tracking methods. Besides, it is worth noting that we evaluate the recent advanced trackers in a new way that defines class-agnostic tracking for tracking unknown object. Finally, ablation studies are provided to analyze the impact of each component and different design choices.

#### A. Experimental Settings

**Datasets:** For 3D single object tracking, we use KITTI [18] and nuScenes [19] datasets for training and evaluation. Since having no access to the ground truth of the official test set of KITTI dataset, we follow P2B [13] and use its public training set to train and evaluate different methods. It contains 8 types of objects in 21 LiDAR sequences, where the scenes 0–16 are for training, scenes 17–18 for validation, and scenes 19–20 for testing. For nuScenes dataset, we use its validation set to evaluate the generalization ability of our model. Note that the nuScenes dataset only annotates key frames, so we report the performance evaluated on the key frames.

**Evaluation metrics:** Following P2B, we use success and precision ratios as metrics for 3D single object tracking. Success calculates the ratio of the intersection over union (IOU) between the prediction and the ground truth bounding box greater than a threshold, and Precision measures distance error between the center points of two bounding boxes from 0–2 m.

**Network architecture:** We sample 1024 points in the search region and 512 points in the template. In the local encoding stage, each point will establish a connection relationship with the points in the 0.3-meter sphere neighborhood. Here we use two layers of GCN [39] for convolution to obtain initialized features of the template and search region. In the feature extraction stage, we set up a three-layers template-aware Transformer module. The output of each layer will be down-sampled by the FPS, and the number of down-sampled points is reduced to 256 for the first layer, and 512 for the second layer. The network
then performs feature aggregation in the search region, gradually
transferring low-density point cloud features to the highest
density. The point-wise features in the above stages have always
maintained 64 dimensions. Note that in this paper, we adopt the
Transformer with multi-head attention having four heads for all experiments.

B. Evaluation on Class-Agnostic Tracking

1) Quantitative Comparison With State-of-The-Art Trackers: A variety of trackers for the 3D SOT task have emerged, each incorporating different approaches within Siamese frameworks. For example, the family of the pointnet++ includes P2B [13], BAT [14], MLVSNet [21], PTTR [38] and V2B [16]; the sparse convolution LTTR [25]; and the Transformer STNet [17]. For a comprehensive comparison, we compare our one-stream framework with these state-of-the-art trackers on the KITTI and nuScenes datasets. To fully illustrate the tracking performance, we evaluate these methods across four categories per dataset. KITTI covers car, pedestrian, van, and bicycle; while nuScenes includes car, pedestrian, truck, and bicycle. It’s important to note that all results of nuScenes are tested by models trained in KITTI.

As shown in the Table I, we summarize and report the performance of these trackers base on the success and precision metrics. Our results are presented in the last row of the table. It can be seen that our method achieves the highest average results across the four categories on KITTI. KITTI covers car, pedestrian, van, and cyclist; while the features in those Siamese frameworks (P2B, BAT, and V2B) are mixed together. This convincingly illustrates that our method makes these two types of features distinguishable, which is beneficial to predicting proposals. On the contrary, poor performances make those Siamese frameworks hard to distinguish the target. Benefiting from it, our method also achieves better results as shown in Table I.

In Fig. 6, we visualize the results of our method (red boxes) and V2B (blue boxes). The first row shows the results of the same scene’s different frames in the car category and the second row shows the results in the pedestrian category. We can see that our method can still identify the target even in the pedestrian dataset full of interference. In contrast, in the case of the interference from similar objects, V2B cannot capture the target. The visualization results illustrate our method has stronger immunity to the interference of similar objects than V2B.

2) Qualitative Comparison With State-of-The-Art Trackers: Our method integrates the feature extraction and relation modeling by a one-stream backbone while the Siamese frameworks establish the relationship after a Siamese backbone. In order to directly show the discrepancy between the above frameworks, we compare the feature distributions of our method (OST) and three representative Siamese frameworks (P2B, BAT, and V2B) in KITTI dataset. Specifically, we use the T-SNE [44] method to visualize the search region features. For a fair comparison, the feature of our OST is visualized after passing both TTM and MFA. As for the Siamese frameworks, we visualize their output features after augmenting by the relation modeling modules. As shown in Fig. 5(a), the point features on the target surface and the background are highlighted in blue and red, respectively. From the last column of this figure, we can see that our OST has obvious boundaries, while the features in those Siamese frameworks (P2B, BAT, and V2B) are mixed together. This convincingly illustrates that our method makes these two types of features distinguishable, which is beneficial to predicting proposals. On the contrary, poor performances make those Siamese frameworks hard to distinguish the target. Benefiting from it, our method also achieves better results as shown in Table I.

C. Evaluation on Class-Agnostic Tracking

1) Settings for Class-Agnostic Tracking in Point Clouds: At this stage, the advanced methods of 3D SOT invariably follow the paradigm that the model is trained in a specific class and tested in this class. However, this mode requires that the

| Methods | Car | Pedestrian | Van | Cyclist | Mean | Mean |
|---------|-----|------------|-----|---------|------|------|
| KITTI   | 6424| 6088       | 1248| 308     | 14068|      |
| nuScenes| 15778| 8019       | 3710| 501     | 2708 |      |

The success/precision are reported and the best results are highlighted in bold.
class to which the tracked object belongs is present in the training dataset. This is a strict hypothesis to achieve in reality, on the contrary, dealing with challenging and out-of-distribution (OOD) scenes in the tracking task requires the model to be able to track any objects. Therefore, we believe that models of advanced single object tracking methods should not only perform well on a specific class but also achieve better results on unseen or OOD classes.

To reflect the advantages of the proposed one-stream method on 3D SOT, we use a class-agnostic setting to evaluate the different methods according to the literature [28]. As shown in Table II, it provides two settings on KITTI dataset. The
categories commonly used for tracking are cars, pedestrians, vans, and cyclists, and we divide the four categories into two groups in two different ways. In setting-1, we divide pedestrians, vans, and cyclists as the group for training and cars as the unseen category for testing. In setting-2, we divide cars, vans, and cyclists as the group for training and pedestrians as the unseen category for testing. In addition, the categories used for training are the ones observed by the model. Under such experimental settings, we can conduct class-agnostic experiments to verify the effectiveness of our method when tested in the unseen category, and the robustness of the model can be measured when tested in the observed category.

2) Results of Class-Agnostic Tracking: To assess the performance of our method in tracking unknown categories, we conduct class-agnostic tracking experiments, including both quantitative and qualitative analyses. We train models on the categories from the train split and subsequently test them on the unseen category and observed categories from the test split.

Quantitative comparisons: The state-of-the-art Siamese trackers (P2B [13], BAT [14], V2B [16]) extract features based on a parameter-shared backbone, then establish the relationship between template and search region by independent relation modeling module. Our OST joints feature extraction and relation modeling by the TTM. To fully compare with those advanced methods when tracking unseen objects, we report the results of the class-agnostic experiment on the KITTI. As shown in Table II, we summarize and report the performance of these trackers according to the success and precision metrics. The unseen category for testing was ‘Car’ in setting-1 and ‘pedestrian’ in setting-2. Our results are shown in the last row of each table. Compared to classical methods, our method, OST, achieved the best results, with a performance improvement of 1.4/2.8 points in setting-1 and 4.5/12.4 points in setting-2. This indicates that our method performs exceptionally well when dealing with unknown objects.

TABLE II

| Split | Category | KITTI               |
|-------|----------|---------------------|
| Train |          | setting-1 | setting-2 |
|       | Pedestrian | Car       | Van       |
|       | Van       | Cyclist    | Cyclist   |
|       | (8123)    | (23045)   |           |
| Test  | Observed | Pedestrian | Car       |
|       | Van       | Cyclist    | Cyclist   |
|       | (7644)    | (7980)    |           |
|       | Unseen    | Car       | Pedestrian|
|       |           | (6424)    | (6088)    |

The success/precision are reported and the best results are highlighted in bold.

Qualitative comparison: In the quantitative analysis above, our method demonstrates superior performance on unseen categories due to the incorporation of template-aware feature extraction. To substantiate this impact, we employ T-SNE [44] to visualize the intermediate features extracted from the testing phrases within the network. For the Siamese-based methods, P2B, BAT and V2B, we visualize the features after relation modeling. While for our method, we directly visualize the features extracted in the one-stream backbone. We visualize the point cloud features of the unseen category object in the search region, as shown in Fig. 5(b). The visualization results of our method are presented on the far right. As can be observed, our method can effectively distinguish target points from background points, even when the target category is unseen during training, as indicated by the green box surrounding the blue points. In contrast, P2B, BAT, and V2B do not exhibit such clear differentiation. We further compare the results obtained from class-specific and class-agnostic experiments within the same scene. In Fig. 8, “Classical” and “Agnostic” mean class-specific setting and class-agnostic setting separately. For class-specific, the tracking objects category is exposed during training while it is not exposed for class-agnostic. As shown in the Fig. 8, we can see that our method can capture the target regardless of whether it has been exposed during training. In contrast, it fails if V2B has not seen such object categories while training. In the bottom two rows of the figure, our method can locate the pedestrian regardless of whether it has been observed during training. V2B can only locate this target when it sees it; if not, it will capture the nearby car, not the target pedestrian.

D. The Computational Cost of Different Methods

For a more comprehensive evaluation of model performance, we compare the proposed OST with P2B [13], BAT [14] and V2B [16] in terms of success, precision, parameters, GFLOPs, FPS, and training time. Specifically, we evaluate our model on the car category in the KITTI dataset and all experiments are on a single TITAN RTX GPU. For training time, we calculate the average time per epoch with batch size 24. As shown in Fig. 7,
To justify the effectiveness of each component, we evaluate these components across four categories in the KITTI dataset covers cars, pedestrians, vans, and cyclists. We report the success and precision in Table IV. We set the local encoding module named local, the template-aware Transformer module named TTM, and the multi-scale feature aggregation module named MFA in the table. It can be seen that adding the TTM improves the tracking performance by 3.4% and 3.0% in car categories, and adding the MFA makes the performance continue to increase by 0.3% and 0.7%. These verify that the network module we have set up is valid. The Transformer layer performs feature extraction and relationship modeling according to the template so that the points in the search region have the information from the template. Multi-scale feature aggregation methods can perceive features at different layers to compensate for the spatial information lost when the network layer is deepened. As a result, adding these modules makes the performance of our method improve.

2) The Effect of the Number of TTM: The number of TTM layers is a key parameter for feature extraction. Here we study the effects of different values of n on tracking accuracy. Fig. 9 shows the network performance provided by different numbers of layers including 0, 1, 2, 3, and 4, respectively. In the figure, the horizontal axis represents the number of layers, and the vertical axis represents the tracking accuracy. Then the green line shows the success our method performed on the car category in the KITTI dataset and the blue line represents the precision. As shown in the Fig. 9, we can see that our tracker achieves the best performance when n = 3 with success 72.0% and precision 84.2%.

3) Comparison of Different Multi-Scale Aggregation Strategies: For feature aggregating, the usual practice is that spread information from deep layer to shallow layer as Fig. 3(b). In our method, we aim to supplement the semantic information with the spatial information by inverse sampling under the premise of ensuring the semantic discrimination of the features. With the idea, we proposed the tracking-specific method that spread information from shallow layer to deep layer. We compare the tracker performance by our setting with common setting. In Table V, we report the results of two settings, the usual practice named OST/w.usual and our method named OST/w.specific. We can observe in the table that our method (OST/w.specific) achieves the better results across the four categories on KITTI. In detail, our method improves by 1.1%/1.3%, 7.2%/7.0%, 5.2%/6.2%, 1.3%/0.4% on the four categories, respectively. The semantic information defines the cues for any given tracking target, so it can help the network to accurately segment the foreground. Therefore, comparing with usual practice, our method can locate the target based on more accurate foreground point information and obtain better tracking performance.

4) Comparison of Siamese Network and One-Stream Network: In addition to the comparison with state-of-the-art Siamese-based trackers, we also disassemble our one-stream framework into Siamese structures for experimental comparison. Specifically, the Transformer module is performed separately instead of splicing the template with the search region, and both branches perform multi-scale feature enhancement. Then we associate the two sides using the relation modeling

### Table IV

| Local | TTM | MFA | Car    | Pedestrian | Van    | Cyclist |
|-------|-----|-----|--------|------------|--------|---------|
| ✓     | ✓   | ✓   | 68.3/80.5 | 47.4/76.2 | 55.0/65.6 | 44.4/54.4 |
| ✓     | ✓   | ✓   | 71.7/83.5 | 48.9/76.7 | 56.4/66.4 | 49.2/60.4 |
| ✓     | ✓   | ✓   | 72.0/84.2 | 51.4/82.6 | 57.5/68.2 | 49.2/60.4 |

Fig. 7. Comparison of GFLOPs, precision, and frames per second (FPS) of different methods. Horizontal and vertical coordinates denote FPS and precision, respectively, and the bubble size indicates GFLOPs. The method that is closer to the upper right corner with a smaller size is better. In addition, the table in this figure shows the success, parameters, and training time.
Fig. 8. Visualization results demonstrate that our methods have powerful generalization abilities. “Classical” and “Agnostic” means class-specific setting and class-agnostic setting separately. The top sub figures are results of a same car sequence tested by models trained in class-specific car category and class-agnostic setting-1; The bottom sub figures are results of a same pedestrian sequence tested by models trained in class-specific pedestrian category and class-agnostic setting-2. The ground truth is plotted in green, and the result boxes are plotted in blue and red for V2B and our method, respectively.

| Method           | Car  | Pedestrian | Van  | Cyclist |
|------------------|------|------------|------|---------|
| Siamese/w.P2B-xcorr | 71.4/83.0 | 48.3/74.7 | 58.7/69.0 | 40.3/54.5 |
| Siamese/w.V2B-xcorr | 71.9/83.2 | 49.2/77.4 | 57.4/64.8 | 40.3/49.0 |
| One-stream       | 72.0/84.2 | 51.4/82.6 | 57.5/68.2 | 49.2/60.4 |

The comparison with siamese networks designed with different methods.

module in P2B and V2B. The results of the above steps are used to predict proposals by the way in our method including segmentation and Voxel-to-Bev head. The experimental results based on the above settings are shown in Table VI. The Siamese framework with P2B relation modeling module is written as Siamese/w.P2B-xcorr and the Siamese framework with BAT relation modeling module is written as Siamese/w.BAT-xcorr. Although Siamese networks have a powerful relation modeling module and their computational complexity is much higher, our OST can also achieve slightly better performance compared with them, especially in cyclist.

5) Comparison of Segmentation and Completion: In our method, we use a 3D detection head provided by V2B [16] for the proposal. Differently, we enhance the features for predicting proposal in another way. V2B uses the completion method to complete the points in the search region and uses the loss to constrain the feature learning in the search region. They aim to enhance the shape information of the points features. Here we use the segmentation method to enhance the features. The purpose is to distinguish the foreground and background points of the search region. In Table VII, we give a comparison of completion and segmentation, and we report the results on KITTI.
The method improves by 2.0%/2.3%, 1.3%/3.6%, 4.5%/5.4%, 8.5%/10.1% on the four categories, respectively.

V. CONCLUSION AND DISCUSSION

In this paper, we proposed a One-stream framework for 3D single object tracking, including two core parts, a template-aware Transformer module (TTM) and a multi-scale feature aggregation module (MFA). The former realizes a novelty of integrating the relation modeling into feature extraction, while the latter reaches a strong alliance between the spatial and semantic information. In this way, our method can efficiently resolve this problem further from the perspective of general encoder (e.g., the asymmetric convolution [34]), capturing the adaptive sequence-specific information (e.g., adaptive and decisive samples [35]), and specific loss (e.g., conditional sharing loss [33]).

REFERENCES

[1] A. I. Comport, É. Marchand, and F. Chaumette, “Robust model-based tracking for robot vision,” in Proc. IEEE/RJSI Int. Conf. Intel1. Robots Syst., 2004, pp. 692–697.
[2] W. Luo, B. Yang, and R. Urtasun, “Fast and furious: Real time end-to-end 3D detection, tracking and motion forecasting with a single convolutional net,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2018, pp. 3569–3577.
[3] E. Machida, M. Cao, T. Murao, and H. Hashimoto, “Human motion tracking of mobile robot with Kinect 3D sensor,” in Proc. IEEE SICE Annu. Conf., 2012, pp. 2207–2211.
[4] M. Kristan et al., “The visual object tracking vot2015 challenge results,” in Proc. IEEE Int. Conf. Comput. Vis. Workshops, 2015, pp. 564–586.
[5] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, “Fully-convolutional siamese networks for object tracking,” in Proc. Eur. Conf. Comput. Vis. Workshops, 2016, pp. 850–865.
[6] M. Kristan et al., “A novel performance evaluation methodology for single-target trackers,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 11, pp. 2137–2155, Nov. 2016.
[7] S. Giancola, J. Zarzar, and B. Ghanem, “Leveraging shape completion for 3D siamese tracking,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 1359–1368.
[8] C. Zheng et al., “Beyond 3D siamese tracking: A motion-centric paradigm for 3D single object tracking in point clouds,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 8111–8120.
[9] S. Tian et al., “Learning the incremental warp for 3D vehicle tracking in LiDAR point clouds,” Remote Sens., vol. 13, 2021, Art. no. 2770.
[10] H. Jiang et al., “Point cloud registration-driven robust feature matching for 3D siamese object tracking,” IEEE Trans. Neural Netw. Learn. Syst., early access, Nov. 13, 2023, doi: 10.1109/TNNLS.2023.3325286.
[11] C. Qi, O. Litany, K. He, and L. J. Guibas, “Deep hough voting for 3D object detection in point clouds,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 9276–9285.
[12] Y. Zhou and O. Tuzel, “VoxelNet: End-to-end learning for point cloud based 3D object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2018, pp. 4490–4499.
[13] H. Qi, C. Feng, Z. CAO, F. Zhao, and Y. Xiao, “P2B: Point-to-box network for 3D object tracking in point clouds,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 6328–6337.
[14] C. Zheng et al., “Box-aware feature enhancement for single object tracking on point clouds,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 13179–13188.
[15] J. Shan, S. Zhou, Z. Fang, and Y. Cui, “PTT: Point-track-transformer module for 3D single object tracking in point clouds,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2021, pp. 1310–1316.
[16] L. Hui et al., “3D Siamese voxel-to-BEV tracker for sparse point clouds,” in Proc. Adv. Neural Inf. Process. Syst., 2021, pp. 28714–28727.
[17] L. Hui et al., “3D Siamese transformer network for single object tracking on point clouds,” in Proc. Eur. Conf. Comput. Vis., 2022, pp. 293–310.
[18] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? The kitti vision benchmark suite,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2012, pp. 3354–3361.
[19] H. Caesar et al., “nuScenes: A multimodal dataset for autonomous driving,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 11618–11628.
[20] C. Qi, L. Yi, H. Su, and L. J. Guibas, “PointNet: Deep hierarchical feature learning on point sets in a metric space,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 5105–5114.
[21] Z. Wang et al., “MLVSNet: Multi-level voting siamese network for 3D visual tracking,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 3081–3090.
[22] Z.-C. Luo et al., “Exploring point-BEV fusion for 3D point cloud object tracking with transformer,” 2022, arXiv:2208.05216.
[23] I. Oleksienko, P. Nousi, N. Passalis, A. Tefas, and A. Iosifidis, “Variational voxel pseudo image tracking,” IEEE Symm. Ser. Comput. Intel., Mexico City, Mexico, pp. 323–328, 2023, doi: 10.1109/SITCS2477.2023.10571810.
[24] J. Zarzar, S. Giancola, and B. Ghahem, “Efficient bird eye view proposals for 3D Siamese tracking,” 2019, arXiv:1903.10368.
[25] Y. Cui, Z. Fang, J. Shan, Z. Gu, and S. Zhou, “3D object tracking with transformer,” in Proc. 32nd Brit. Mach. Vis. Conf., Nov. 22–25, 2021, p. 317.
Xiantong Zhao is currently working toward the Ph.D. degree with the School of Mathematical Sciences, Dalian University of Technology, Dalian, China. His research interests include computer graphics and 3D point cloud processing.

Yinan Han is currently working toward the master’s degree, with the School of Mathematical Sciences, Dalian University of Technology, Dalian, China. His research focuses on 3D point cloud processing.

Shengjing Tian received the Ph.D. degree from the School of Mathematical Sciences, Dalian University of Technology, Dalian, China. He is currently a Postdoctoral Teaching Fellow with the China University of Mining and Technology, Xuzhou, China. His research interests include multimodal learning, point cloud understanding, and visual tracking. The related research works have been published in many important journals including IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON MULTIMEDIA, and IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHIC. He is also a Reviewer for various journals, such as the Visual Computer, Information Fusion, and IEEE TRANSACTIONS ON MULTIMEDIA.

Jian Liu received the Ph.D. degree from the Shandong University, China, in 2022. He is currently a Postdoctor with the School of Software, Tsinghua University, Beijing, China. He has authored papers in international journals and conferences, including TOG, Computer-Aided Design, Computer Aided Geometric Design, ICCV, and ICRA. His research interests include computer graphics, computer vision, robotic grasping, and machine learning.

Xiuping Liu She received the M.S. degree from Jilin University, Changchun, China and Ph.D. degree from the Dalian University of Technology, Dalian, China. She is currently a Professor with the School of Mathematical Sciences, Dalian University of Technology. Her research interests include computer vision, computer graphics, and machine learning. She has published more than 70 papers in international journals and conferences, including TOG, IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, IEEE TRANSACTIONS ON IMAGE PROCESSING, ECCV, and ICCV.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.