1. More Implementation Details

1.1. Model Details

We adopt DGCNN [8] as the backbone network to extract local geometric information, which contains 3 EdgeConv layers and 3 downsampling layers. In the target-centric transformer, all shared MLPs for targetness mask prediction and target center regression have 4 linear layers, each of which is followed by BatchNorm [5] and ReLU [1].

1.2. Training & Inference

We train our model using the Adam optimizer [6] with an initial learning rate of $10^{-3}$. The learning rate is reduced to $1/5$ every 40 epochs. The batch size is empirically set to 128. During inference, the model tracks the target frame-by-frame using the previous predicted bounding box, with the bounding box of the first frame known as ground truth.

1.3. Data Augmentation

For the SOT task, the network only needs to consider a sub-region of the whole scene where the tracking target may appear. For training, we enlarge the ground truth bounding box by 2 meters to obtain the sub-region. We then sample 1024 points inside the region to generate the input point clouds $P_{t-1}$ and $P_t$. To simulate the inaccurate predictions during inference, we augment the input 3D bounding box $B_{t-1}$ by performing random translation with a range of $[-0.3m, 0.3m]$ in all directions as well as random rotation around the up-axis.

2. Ablation Study

2.1. Numbers of Layers and Heads

To explore the impact of the number of layers $N_L$ and the number of attention heads $h$, we conduct experiments and report the results in Tab. 1. We observe no significant performance gains from using more than 4 layers and 1 head, but at the cost of more parameters and lower inference speed. Thus we set $N_L = 4$ and $h = 1$ for higher efficiency and better performance.

2.2. Hyper-parameter Selection

We change one hyper-parameter each time and fix others for fair comparison. As shown in Tab. 2, except for $\gamma_2 = 1.0$, CXTrack only has a minor improvement compared with other settings, which demonstrates the robustness to hyper-parameter selection. X-RPN distinguishes points of interest using target center predictions. In the case $\gamma_2 = 1.0$, weak supervisory signal on center prediction degrades the performance of CXTrack.

3. More Analysis

3.1. Comparison with M2-Track

To make a fair comparison with previous state-of-the-art method, M2-Track [9], we follow the experimental set-
Table 3. Comparison with M2-Track on nuScenes. We both train and test various methods on nuScenes.

<table>
<thead>
<tr>
<th>Category</th>
<th>Method</th>
<th>Dense</th>
<th>Medium</th>
<th>Sparse</th>
</tr>
</thead>
<tbody>
<tr>
<td>Car(64159)</td>
<td>M2-Track</td>
<td>65.9/73.4</td>
<td>53.4/62.0</td>
<td>42.3/53.5</td>
</tr>
<tr>
<td></td>
<td>CXTrack</td>
<td>52.4/57.4</td>
<td>44.4/51.6</td>
<td>42.9/48.4</td>
</tr>
<tr>
<td>Pedestrian(33227)</td>
<td>M2-Track</td>
<td>39.2/68.2</td>
<td>35.9/66.4</td>
<td>17.2/37.1</td>
</tr>
<tr>
<td></td>
<td>CXTrack</td>
<td>41.1/69.4</td>
<td>34.8/62.8</td>
<td>19.8/37.2</td>
</tr>
<tr>
<td>Truck(13587)</td>
<td>M2-Track</td>
<td>67.3/68.9</td>
<td>57.3/57.4</td>
<td>46.6/46.8</td>
</tr>
<tr>
<td></td>
<td>CXTrack</td>
<td>54.3/53.6</td>
<td>48.7/48.7</td>
<td>42.6/41.3</td>
</tr>
<tr>
<td>Trailer(3352)</td>
<td>M2-Track</td>
<td>54.2/52.5</td>
<td>61.3/63.7</td>
<td>57.6/59.6</td>
</tr>
<tr>
<td></td>
<td>CXTrack</td>
<td>59.0/52.9</td>
<td>61.3/57.2</td>
<td>56.6/50.9</td>
</tr>
</tbody>
</table>

Figure 1. Sometimes background points also make reasonable prediction. Red indicates the predicted target centers of all points, including both foreground points and background points.

settings in M2-Track and train our proposed CXTrack from scratch on nuScenes [2]. NuScenes poses a greater challenge than KITTI due to its lower frequency for annotated frame, which enlarges the motion vectors of tracked targets between two consecutive frames and leads to larger appearance variation. Similar to LiDAR-SOT, we split NuScenes into three sets according to the sparsity of the tracking target. As shown in Tab. 3, CXTrack performs better than M2-Track on dense point clouds and low-speed tracked targets such as pedestrians and trailers, but fails to tackle sparse point clouds or high-speed targets. Compared with M2-Track (based on PointNet [7]) which overlooks local geometry information, CXTrack adopts a hierarchical network to extract point features, thereby relying more on the quality of point clouds and achieving limited performance under sparse scenes.

3.2. Comparison with Voxel-based localization head

Compared to ST-Net [4], which employs a voxel-based localization head [3], our proposed point-based CXTrack exhibits a minor decrement in performance for the Car category, while outperforming ST-Net on other categories. We argue that the performance drop stems from the following reasons. First, we observe that most vehicles have simple shapes and large size, which fit well in voxels. Voxelization in ST-Net can provide a strong shape prior, thereby leading to more precise localization than point-based CXTrack.

Secondly, our proposed X-RPN relies on target center predictions to distinguish points of interest from the background points. However, if some background points also make similar center predictions, as shown in Fig. 1, X-RPN may introduce noise into the point-feature interaction. Conversely, the limited receptive field of 3D CNN utilized in voxel-based head restrict point-feature interaction to a local region of 3D space, thereby distinguishing points of interest from background. The lack of distractors for cars also makes our improvement over previous methods insignificant. Nevertheless, for other categories, such as pedestrians, that have complex shapes and small sizes, voxelization introduces considerable information loss, which significantly degrades the tracking performance.

3.3. Visualization of the activation maps in X-RPN

Some example attention maps of the local transformer in X-RPN are shown in Fig. 2. We find that CXTrack attends to the tracked targets of all categories correctly even with intra-class distractors (Pedestrian in Fig. 2), owing to the center embedding.

References


