Supplementary Material for "CXTrack: Improving 3D Point Cloud Tracking with Contextual Information"

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

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 $\mathcal{P}_{t-1}$ and $\mathcal{P}_t$. To simulate the inaccurate predictions during inference, we augment the input 3D bounding box $\mathcal{B}_{t-1}$ 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.

Table 1. **Ablation studies of the number of layers and the number of attention heads on KITTI.** N_L is the number of transformer layers. h denotes the number of attention heads.

N_L	h	Car	Pedestrian	Van	Cyclist	Mean
3	1	67.1/79.3	64.4/90.5	53.9/64.5	72.0/93.5	64.9/83.1
4	1	69.1/**81.6**	**67.0/91.5**	**60.0/71.8**	**74.2/94.3**	**67.5/85.3**
5	1	68.8/79.7	65.2/89.9	59.4/70.9	73.6/**94.3**	66.5/83.7
3	2	68.7/80.2	65.0/90.4	55.3/65.1	72.9/93.7	66.0/83.6
4	2	**69.4**/80.5	65.0/89.6	57.1/71.2	73.5/94.1	66.5/83.9

2.2. Hyper-parameter Selection

Table 2. **Ablation studies of different hyper-parameters on KITTI.**

$\frac{1}{\sigma^2}$	γ_1	γ_2	γ_3	Pedestrian
0.1	0.2	10.0	1.0	**67.0/91.5**
0.07				63.5/89.5
0.13				65.9/89.7
	0.1			65.3/90.6
	0.3			66.0/90.3
		1.0		59.4/87.6
		20.0		65.4/90.4
			0.5	66.3/90.8
			2.0	65.5/90.2

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-

*corresponding author

Table 3. **Comparison with M2-Track on nuScenes.** We both train and test various methods on nuScenes.

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

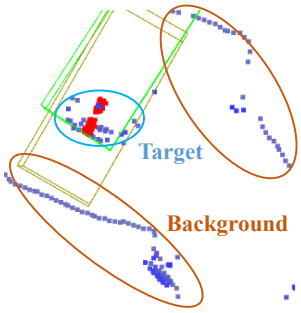

Figure 1. **Sometimes background points also make reasonable prediction.** Red indicates the predicted target centers of all points, including both foreground points and background points.

tings 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 featuers, 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.

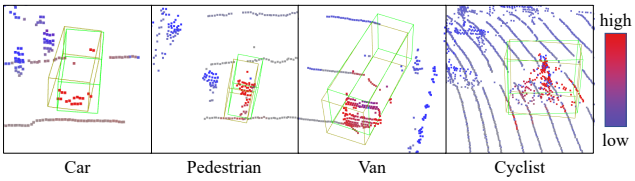

Figure 2. **Representative examples of attention maps in X-RPN.** Red indicates points with higher attention scores.

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

[1] Abien Fred Agarap. Deep learning using rectified linear units (ReLU). *arXiv preprint arXiv:1803.08375*, 2018. 1

[2] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020. 2

[3] Le Hui, Lingpeng Wang, Mingmei Cheng, Jin Xie, and Jian Yang. 3D Siamese voxel-to-BEV tracker for sparse point clouds. *Advances in Neural Information Processing Systems*, 34:28714–28727, 2021. 2

[4] Le Hui, Lingpeng Wang, Linghua Tang, Kaihao Lan, Jin Xie, and Jian Yang. 3D Siamese transformer network for single object tracking on point clouds. *arXiv preprint arXiv:2207.11995*, 2022. 2

[5] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015. 1

[6] Diederik P Kingma and Jimmy Ba. ADAM: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[7] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017. 2

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

[9] Chaoda Zheng, Xu Yan, Haiming Zhang, Baoyuan Wang, Shenghui Cheng, Shuguang Cui, and Zhen Li. Beyond 3D Siamese tracking: A motion-centric paradigm for 3D single object tracking in point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8111–8120, 2022. 1