Supplementary Material
Boosting 3D Single Object Tracking with 2D Matching Distillation and 3D Pre-training

Qiangqiang Wu[1], Yan Xia*[2], Jia Wan[3], and Antoni B. Chan[1]

[1] Department of Computer Science, City University of Hong Kong
[2] School of Engineering and Design, Technical University of Munich
[3] School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen
qiangqwu2-c@my.cityu.edu.hk, yan.xia@tum.de, jiawan1998@gmail.com, abchan@cityu.edu.hk

In this supplementary material, we provide additional details on implementation, overall architecture, ablation study, and qualitative visualization. Sec. A illustrates the implementation details in our Target-Aware Projection (TAP) module and 3D trackers (i.e., SiamDisst and MemDisst). Sec. B details the overall architecture of our proposed 3D trackers including both SiamDIsst and MemDisst. More ablation studies are conducted in Sec. C. More tacking results w/ limited training data is shown in Sec. D. The additional results is presented in 5. Sec. F shows the qualitative visualization of the 3D tracking results and projected 2D images w/ or w/o target-aware interaction generated by our approach. We discuss the limitation and future work in Sec. G.

A Implementation Details

For the TAP module, it consists of a backbone network, a lightweight cross-attention module $\mathcal{C}$ and a CNN-based head g. The backbone network and the head g follow the same implementation in [8]. The lightweight cross-attention module $\mathcal{C}$ is implemented as a simple transformer layer with 2 heads and 96 hidden dimensions. The matching module is used to extract target-aware search features, which are further input to the voxel-based head in SiamDisst or point cloud (PC) based head in MemDisst for 3D box prediction. Note that the memory mechanisms used in STNet [3] and MbpTrack [9] are different, i.e., STNet mainly focuses on sampling memory points from the initial template frame and the previous frame, while MBPTrack samples memory points from three historic frames for matching. Without any modifications, we follow the same memory sample mechanisms in STNet and MBPTrack to sample the memory points for training our TAP modules used in SiamDisst and MemDisst, respectively. The hyper-parameter λ is set to 1.0 for MemDisst and 0.1 for SiamDisst.

Following one pass evaluation [5], we use *Precision* and *Success* metrics to evaluate the 3D tracking performance. Specifically, the *Precision* metric calculates the AUC for the distance between the centers of the predicted 3D bounding

* Corresponding Author

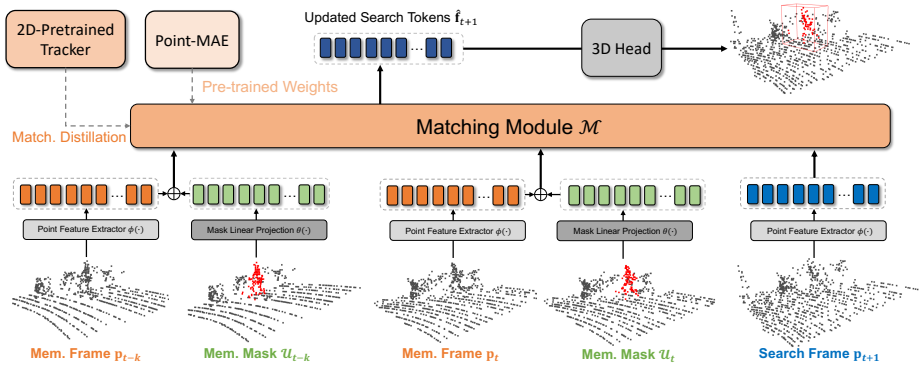

Fig. 1: The overall architecture of the proposed MemDisst. Note that SiamDisst shares a similar overall architecture with MemDisst, except that it drops the mask linear projection $\theta(\cdot)$ since the template used in SiamDisst is well cropped. The dashed and solid lines indicate the steps used in the training and inference stages, respectively.

Variants	Learnable TAP	Unlearnable TAP	IoU Weighting	succ./prec.
SiamBase	✓		✓	**63.5**/**75.0**
SiamBase	✓			56.4/66.3
SiamBase		✓	✓	61.1/72.3
SIamBase		✓		56.0/65.1

Table 1: Comparisons of our SiamBase with various strategies.

box and the ground-truth box from 0 to 2 meters. The *Success* metric evaluates the IOU between the two 3D bounding boxes. All datasets use the same *Precision* and *Success* metrics for evaluation.

B Architecture

The overall architecture of our 3D trackers (i.e., SiamDisst and MemDisst) is shown in Fig. 1. The memory frames can be constructed with a well-cropped template PC in SiamDisst or contextual memory frames in MemDisst. The matching distillation and initialization with 3D pre-trained weights are denoted as dashed lines, which are only used during the training stage. The proposed SiamDisst and MemDisst can run efficiently at 25 and 90 FPS on a single RTX-3090 GPU, respectively.

C More Ablation Studies

Unlearnable Proj. w/o IoU weighting. In Table 1, following the same comparison in the main paper, for unlearnable TAP, we remove the learnable projection module in the TAP and use the z-axis value as the projected feature in

Distillation Method	λ	succ./prec.
IoU-guided Match. Distillation	0.1	**63.5**/**75.0**
Feat.-based Distillation	0.01	59.41/70.11
Feat.-based Distillation	0.1	**61.9**/**72.6**
Feat.-based Distillation	0.5	58.6/69.9
Feat.-based Distillation	1.0	56.2/67.1

Table 2: Comparisons of our IoU-guided matching distillation with feature-based distillation on SiamDisst.

Variant	3D Pre-train.	2D Distill.	Car	Pedestrian	Van	Cyclist
SiamBase	✓		72.5/83.7	52.7/80.1	60.3 69.3	73.9/94.2
SiamBase	✓	✓	**73.7**/**85.1**	**58.4**/**83.9**	**63.5**/**75.0**	**76.9**/**94.5**
MemBase	✓		73.1/85.2	68.2/93.6	63.5/76.5	75.8/94.2
MemBase	✓	✓	**74.1**/**85.6**	**69.1**/**94.1**	**66.6**/**79.3**	**77.2**/**94.7**

Table 3: Component analysis of SiamDisst and MemDisst on the KITTI dataset [2].

the 2D image plane for each of the 3 image channels. The unlearnable TAP w/ IoU weighting obtains inferior performance to our full variant (learnable TAP w/ IoU weighting), which is mainly because the projected images are not good enough to provide more reliable 2D features for distillation. Without using the IoU weighting, the performance is severely degraded.

Feature-based distillation. Similar to the previous feature-based distillation [1, 4, 7], we use the 2D template and search features to directly supervise the 3D template and search features, which can be formulated as:

$$\mathcal{L} = \frac{\lambda}{2}(\mathcal{L}_1(H(\hat{\mathbf{f}}_t), \tilde{\mathbf{f}}_t) + \mathcal{L}_1(H(\hat{\mathbf{f}}_m), \tilde{\mathbf{f}}_m)), \tag{1}$$

where λ is a scaling hyper-parameter and is also used in our loss (Eq. (6) in Main paper), and $H(\cdot)$ is a learnable linear projection to align the 3D dimension to be consistent with the 2D dimension. We test various choices of λ for the feature-based distillation in Table 2 for a fair comparison. Our IoU-guided matching distillation outperforms the optimal feature-based distillation with a margin of 1.6%/2.4% on KITTI-Van. Moreover, our IoU-guided matching distillation has no need to learn an additional linear projection $H(\cdot)$ for dimension alignment.

Component analysis. In Table 3, we show the component analysis results on all categories of the KITTI dataset [2]. With the usage of 2D matching distillation, our approaches obtain consistent improvements over all the categories, which demonstrates the effectiveness of our matching knowledge transfer.

D Learning in the Low-Data Regime

We test the robustness of our SiamDisst in the low-data regime. We subsample the original KITTI-Van training set into various subset percentages. The TAP

Training subset percentage	15%	30%	60%	100%
No. training frames.	299	598	1196	1994
baseline, random	31.7/35.7	40.7/43.1	53.4/60.0	55.6/64.3
+ 3D pre-training	33.1/36.8	42.8/46.5	54.6/64.3	60.3/69.3
+ 2D distillation (SiamDisst)	**37.1**/**40.3**	**46.5**/**50.6**	**57.7**/**66.4**	**63.5**/**75.0**

Table 4: Comparisons of variants trained with different training percentages on KITTI-Van. The best performance (success/precision) is in bold. Our full variant SiamDisst achieves favorable performance in the low-data regime (e.g., 15% and 30%).

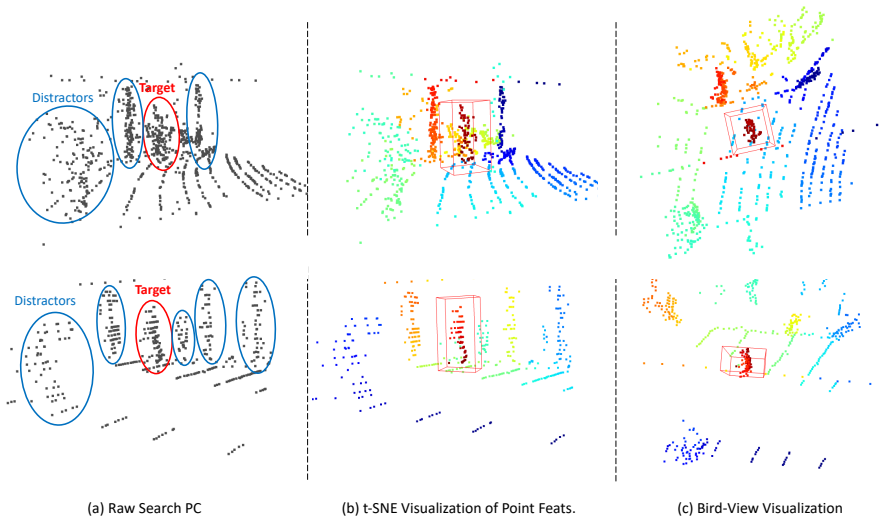

Fig. 2: We input the (a) raw search PC from KITTI-Pedestrian to the matching module of SiamDisst to extract target-aware search features, which are visualized in (b) and (c). The points with a similar color mean that their point features are close in the feature space based on t-SNE. The *red bbox* in (b) and (c) is predicted by our DiamDisst.

module is re-trained in each training situation for fair evaluation. We use the original testing set in KITTI-Van for evaluating all variants. Table 4 shows that by adding the 2D matching distillation, our approach achieves large performance gains even for low amounts training data.

E More Results

We follow the evaluation in MBPTrack and train our models with the larger dataset nuScenes (see Table 5). MemDisst outperforms MBPTrack, and ours is lightweight (6.22M), runs faster (89FPS) than MBPTrack (60FPS, 7.38M) and M2-Track (57FPS). Notably, MemDisst improves a lot in 'Trailer', showing its potential as an effective baseline for learning with limited data. In addition, fpr the speed comparison, memDisst is more lightweight than MBPTrack (6.22M

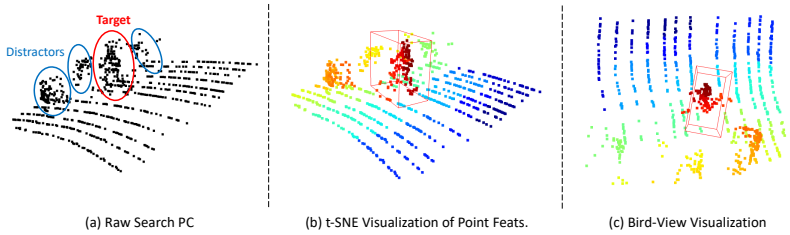

Fig. 3: We input the (a) raw search PC from KITTI-Cyclist to the matching module of SiamDisst to extract target-aware search features, which are visualized in (b) and (c). The points with the similar color mean that their point features are close in the feature space based on t-SNE. The *red bbox* in (b) and (c) is predicted by our DiamDisst.

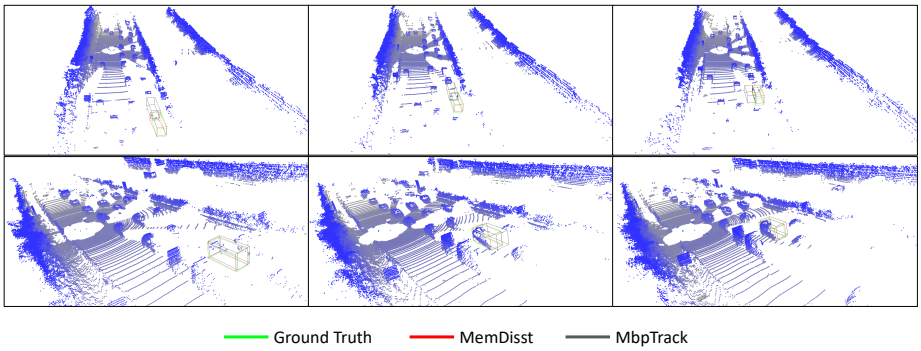

Fig. 4: Visualization of the qualitative tracking results on the sparse scenes (i.e., KITTI-Van). Best viewed on a screen.

vs 7.38M) and uses a more efficient backbone (PointNet vs DGCNN), making it significantly faster. For SiamDisst, it uses the same hierarchical backbone in STNet and extracts dense points. The global attention in SiamDisst increases the complexity, thus limiting the speed (25fps).

F Qualitative Visualization

Tracking results. We visualize the qualitative tracking results in Fig. 4. We compare our MemDisst with the contextual memory-based MBPTrack. As can be seen, our MemDisst can effectively handle sparse LiDAR scans and more accurately predict the 3D bounding box. This can explain why our MemDisst can significantly outperform MbpTrack with a large margin of 5.3%/6.6% on KITTI-Van.

t-SNE visualization. We visualize the extracted updated search features in Fig. 2 and Fig. 3 using t-SNE [6]. The examples are sampled from KITTI-

Method	FPS	Car(64159)	Ped.(33227)	Truck(13587)	Trailer(3352)
P2B	46	38.8/43.2	28.4/52.2	43.0/41.6	49.0/40.1
BAT	65	40.73/43.29	28.83/53.32	45.34/42.58	52.59/44.89
M2-Track	57	55.85/65.09	32.10/60.92	57.36/59.54	57.61/58.26
MBPTrack	60	62.5/70.4	45.3/74.0	62.2/63.3	65.1/61.3
MemDisst	**89**	**63.3/71.5**	**46.6/74.9**	**63.5/64.7**	**67.5/63.4**

Table 5: Comparison on the nuScenes dataset.

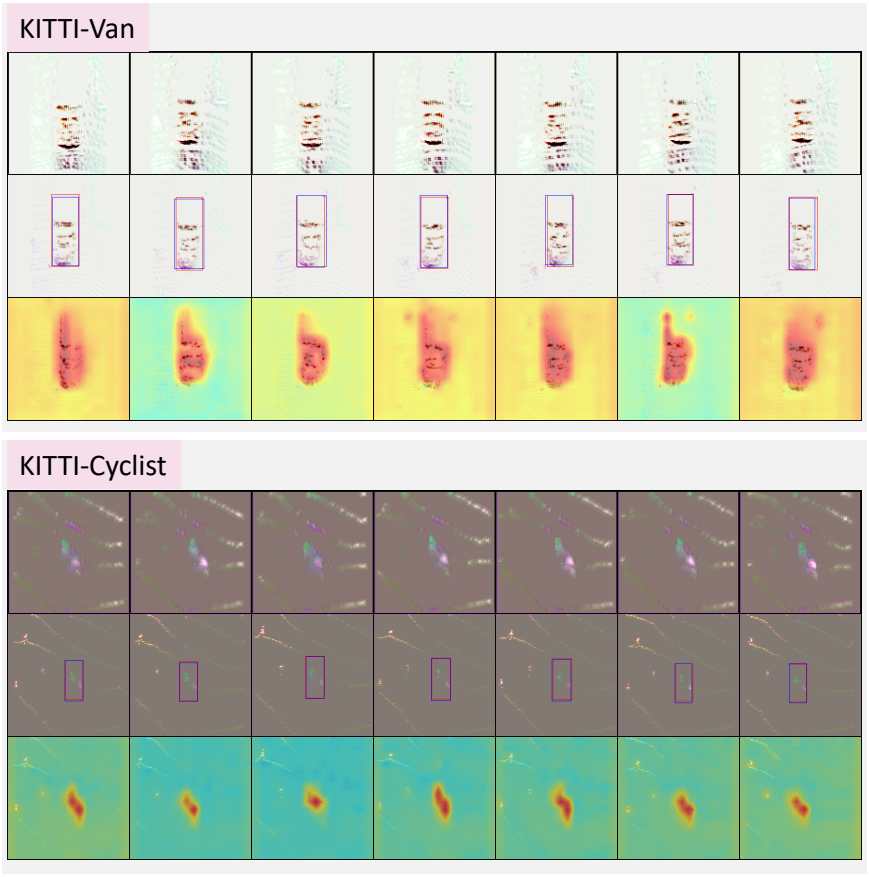

Fig. 5: Visualization of the projected 2D contextual memory frames (i.e., the top row), search frames (middle) and average attention maps (bottom) obtained by MemDisst. The red box indicates the ground-truth (GT) 2D box projected from the GT 3D box. The blue box denotes the prediction of the 2D pre-trained OSTrack [10].

Pedestrian and KITTI-Cyclist, which contains various distractors during the online tracking. The search features extracted by our SiamDisst can well identify the target and distractors, which well supports the following 3D bbox prediction.
Visualization of the projected 2D search frames, contextual memory frames and attention maps. In Fig. 5, we sample memory and search frames from $(t-1)$-th frame to $(t+5)$-th frame and t-th frame to $(t+6)$-th frame for visualization, respectively. For the (t-th) prediction in the search frame, the previous k memory frames are used for matching, and the average attention maps are shown in the bottom. Our TAP module can effectively project 3D point cloud into 2D space, which leads to accurate 2D prediction obtained by the 2D pre-trained OSTrack.

G Limitation

In this paper, we improve the power of 3D point cloud tracking via 3D pre-training and 2D matching distillation. We demonstrate that a simple yet effective target-aware projection (TAP) module can be used to effectively bridge the gap between 2D and 3D tracking. However, this TAP module is still trained separately in the first stage of training. Although it is lightweight (0.49MB) and the training is efficient, it is still not convenient enough to perform the two-stage training on all the datasets. One naive solution is that we train the TAP module jointly with the 3D tracker in the 3D training stage, but this may lead to a trivial solution since the TAP module in the 2D teacher may easily adapt to the 3D tracker via the joint learning, rather than the 3D tracker is guided by the 2D teacher. In the future, we aim to solve this limitation and provide an end-to-end learnable solution.

References

1. Ben-Baruch, E., Karklinsky, M., Biton, Y., Ben-Cohen, A., Lawen, H., Zamir, N.: It's all in the head: Representation knowledge distillation through classifier sharing. In: arXiv:2201.06945 (2022) 3
2. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The International Journal of Robotics Research **32**(11), 1231–1237 (2013) 3
3. Hui, L., Wang, L., Tang, L., Lan, K., Xie, J., Yang, J.: 3d siamese transformer network for single object tracking on point clouds. In: European Conference on Computer Vision. pp. 293–310. Springer (2022) 1
4. Jin, X., Peng, B., Wu, Y., Liu, Y., Liu, J.: Knowledge distillation via route constrained optimization. In: IEEE/CVF International Conference on Computer Vision (2019) 3
5. Kristan, M., Matas, J., Leonardis, A., Vojir, T., Pflugfelder, R., Fernandez, G., Nebehay, G., Porikli, F., Cehovin, L.: A novel performance evaluation methodology for single-target trackers. IEEE Transactions on Pattern Analysis and Machine Intelligence (2016) 1

6. der Maaten, V., Laurens, Hinton, G.: Visualizing data using t-sne. In: Journal of machine learning research (2008) 5
7. Romero, A., Ballas, N.: Fitnets: Hints for thin deep nets. In: arXiv:1412.6550 (2014) 3
8. Wang, Z., Yu, X., Rao, Y., Zhou, J., Lu, J.: P2p: Tuning pre-trained image models for point cloud analysis with point-to-pixel prompting. Advances in neural information processing systems **35**, 14388–14402 (2022) 1
9. Xu, T.X., Guo, Y.C., Lai, Y.K., Zhang, S.H.: Mbptrack: Improving 3d point cloud tracking with memory networks and box priors. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9911–9920 (October 2023) 1
10. Ye, B., Chang, H., Ma, B., Shan, S., Chen, X.: Joint feature learning and relation modeling for tracking: A one-stream framework. In: European Conference on Computer Vision. pp. 341–357 (2022) 6