# Appendix of TrackAny3D

## 1. More Implementation Details

**Datasets.** We conduct comprehensive experiments using three widely-used datasets: KITTI [2], NuScenes [1], and Waymo Open Dataset (WOD) [4]. The KITTI dataset consists of 21 training sequences and 29 test sequences. Due to the unavailability of test labels, we follow previous work [3, 5] and split the training dataset into three subsets: sequences 0-16 for training, sequences 17-18 for validation, and sequences 19-20 for testing. Compared to KITTI, NuScenes and WOD are more challenging. NuScenes contains 700 scenes for training, 150 for validation, and 150 for testing. WOD includes 1121 trajectories, which are categorized into easy, medium, and hard subsets based on the sparsity of point clouds.

**Evaluation Metrics.** We evaluate tracking performance using the One Pass Evaluation (OPE) with success and precision metrics. Success is calculated as the Intersection over Union (IoU) between the predicted bounding box and the ground truth bounding box, while precision measures the distance between the centers of the two corresponding bounding boxes.

**Training and Testing.** We train our model for 160 epochs in KITTI and 60 epochs in NuScenes with a batch size of 32. The Adam optimizer is adopted with an initial learning rate of 0.001. The learning rate is reduced to 0.2 times its current value every 40 epochs (every 10 epochs for nuScenes). For the loss function, we calculate the Mean Squared Error (MSE) between the predicted offset and the ground truth offset. Besides, to normalize the loss based on object size, we divide the MSE by $whl$, which represents the size of the bounding box ($w$ is for width, $h$ is for height and $l$ is for length) for each object.

## 2. More Experimental Analyses

**Bottleneck of MoGE and Adapter.** In Table 1 and Table 2, we investigate the impact of varying bottlenecks for adapters and MoGE across different configurations, respectively. It can be observed that an excessively large bottleneck leads to over-parameterization, increasing optimization challenges and resulting in overfitting. Conversely, too small a bottleneck restricts the model's ability to learn complex features effectively. For adapters, a bottleneck size of 72 offers the best balance, while for MoGE, setting the bottleneck size to 1/8 of the Transformer dimension provides the optimal configuration. This is the final configuration we adopted.

Table 1. Ablations for bottleneck of adapters(AD).

| Bottleneck | TP(M) | Car | Pedestrian | Van | Cyclist | Mean |
|---|---|---|---|---|---|---|
| 192 | 7.6 | 70.1/81.2 | 48.6/75.1 | 70.0/81.7 | 73.9/93.2 | 60.9/78.9 |
| 128 | 6.4 | 70.4/81.2 | 56.7/82.3 | 72.9/84.6 | 70.9/92.2 | 64.7/82.2 |
| 72 | 5.3 | 73.4/85.2 | 59.6/85.6 | 70.0/82.8 | 74.7/94.0 | 67.1/85.4 |
| 48 | 4.9 | 70.6/83.8 | 56.2/84.8 | 69.0/80.0 | 63.2/90.4 | 64.1/84.0 |

Table 2. Ablations for bottleneck of MoGE.

| Bottleneck | TP(M) | Car | Pedestrian | Van | Cyclist | Mean |
|---|---|---|---|---|---|---|
| /2 | 10.7 | 70.7/82.3 | 55.7/83.4 | 70.5/82.9 | 71.4/92.5 | 64.2/83.1 |
| /4 | 7.1 | 70.2/81.5 | 53.0/79.2 | 70.1/82.7 | 73.7/93.7 | 62.8/80.9 |
| /8 | 5.3 | 73.4/85.2 | 59.6/85.6 | 70.0/82.8 | 74.7/94.0 | 67.1/85.4 |
| /16 | 4.5 | 71.9/83.9 | 53.1/82.3 | 70.8/83.1 | 72.1/93.2 | 63.7/83.3 |

Table 3. Ablations for the numbers of input point clouds of the template and search regions. "$N_t$" and "$N_s$" denote the point numbers of template and search region, respectively.

| Point Nums | Car | Pedestrian | Van | Cyclist | Mean |
|---|---|---|---|---|---|
| $N_t$=128, $N_s$=128 | 73.4/85.2 | 59.6/85.6 | 70.0/82.8 | 74.7/94.0 | 67.1/85.4 |
| $N_t$=128, $N_s$=256 | 70.2/81.8 | 56.3/82.3 | 71.5/82.8 | 71.7/92.6 | 64.3/82.3 |
| $N_t$=256, $N_s$=256 | 69.2/81.5 | 60.8/86.4 | 67.8/79.7 | 74.4/93.5 | 65.6/83.7 |

Table 4. Ablations for the number of all (M) and selected experts (K) in MoGE. "TP" denotes Tunable Parameters.

| | TP(M) | Car | Pedestrian | Van | Cyclist | Mean |
|---|---|---|---|---|---|---|
| M=8, K=4 | 5.3 | 73.4/85.2 | 59.6/85.6 | 70.0/82.8 | 74.7/94.0 | 67.1/85.4 |
| $M$=8,$K$=2 | 5.3 | 72.3/83.4 | 57.6/84.0 | 69.0/80.9 | 71.2/92.6 | 65.6/83.6 |
| $M$=4,$K$=2 | 4.4 | 71.9/83.4 | 59.2/86.3 | 71.9/83.9 | 73.5/93.4 | 66.4/84.9 |
| $M$=4,$K$=1 | 4.4 | 70.6/82.2 | 61.6/87.9 | 68.5/85.3 | 76.2/94.2 | 66.7/83.7 |

**Numbers of Input Point Clouds.** Table 3 shows an analysis on the sample input point numbers of template and search regions. We noticed that increasing the point number of regions does not effectively improve performance. It may introduce more background noise for regions. Considering both effectiveness and inference speed, we have selected $N_s$=128 and $N_t$=128 as our final configuration.

**Setting of MoGE.** In Table 4, we conduct an analysis of the number of all selected experts for MoGE. Our findings indicate that simply increasing the total number of experts ($M$) and the number of selected experts ($K$) does not lead to a linear improvement in overall performance. In fact, continuously increasing the total number of experts increases the learning burden. We ultimately chose $M$=8 and $K$=4. This configuration was applied in all our experiments.

**Parameter Control in Category-Unified Experiments.** We take MBPTrack as an example and add an experiment in which we adjust its architecture to match the total model parameter count (TM) of TrackAny3D. Specifically, we expand MBPTrack's feature dimensions in its backbone and increase the number of its Transformer layers to 12, resulting in a total of 26.3M parameters. As shown in Table 5, we can observe that simply increasing the total number of parameters does not necessarily lead to optimal performance, and TrackAny3D still outperforms MBPTrack under similar total parameter budgets.

Furthermore, one of the key advantages of our approach is its ability to efficiently transfer large pre-trained models to category-agnostic 3D single object tracking (SOT), using only a small number of tunable parameters (TP). Our method requires just 5.3M tunable parameters, which is significantly fewer than existing methods such as CXTrack (18.3M) and MBPTrack (7.4M). This aligns with recent trends in 2D vision and natural language processing (NLP), where efficient transfer of large models has become a central research focus. Comparisons based on the number of tunable parameters have increasingly served as a standard evaluation criterion for assessing model efficiency and transferability.

**Ablation Analysis of MoGE Architecture.** As shown in Tab. 6, we conducted two additional experiments: one removes the router (i.e., all experts are merged through direct summation), and the other compares MoGE with a capacity-matched dense MLP under the same parameter budget. The results show that our MoGE still outperforms these two counterparts, indicating that the performance gains stem from the design of MoGE, including the routing mechanism and the use of multiple experts, rather than merely increasing the number of parameters.

**Comparison of Pretraining Strategies.** We compare various full fine-tuning configurations, including fine-tuning from random initialization and fine-tuning from pretrained RECON weights. Tab. 7 consistently shows that both fine-tuning settings achieve lower performance compared to our method. This is because the pretrained RECON model retains strong general knowledge and feature extraction capabilities that are beneficial for downstream tasks, whereas full fine-tuning on these limited downstream datasets risks catastrophic forgetting of the pretrained knowledge, leading to a degradation in performance.

**Impacts of Pretrained Model.** We further investigate the impact of different pretrained models on TrackAny3D by replacing an additional pretrained model, PointMAE. As shown in Tab. 8, initializing with PointMAE also achieves strong performance, demonstrating the broad effectiveness of the paradigm we initially proposed for transferring pretrained models to 3D SOT is viable, and our migration method is indeed effective.

Table 5. Ablations for Total Model Parameters (TM).

| Method | TP(M) | TM(M) | Car | Pedestrain | Van | Cyclist | Mean |
|---|---|---|---|---|---|---|---|
| CXTrack | 18.3 | 18.3 | 60.2/72.6 | 54.6/81.6 | 57.6/70.0 | 44.4/57.0 | 57.2/75.9 |
| MBPTrack | 7.4 | 7.4 | 62.3/72.1 | 50.2/80.9 | 66.6/78.2 | 71.8/92.2 | 56.1/74.9 |
| MBPTrack* | 26.3 | 26.3 | 63.1/77.1 | 46.0/74.6 | 67.6/77.4 | 66.7/92.0 | 56.2/76.4 |
| TrackAny3D | 5.3 | 27.2 | 73.4/85.2 | 59.6/85.6 | 70.0/82.8 | 74.7/94.0 | 67.1/85.4 |

Table 6. Ablations on the MoGE Architecture.

| Methods | TP(M) | TM(M) | Car | Pedestrian | Van | Cyclist | Mean |
|---|---|---|---|---|---|---|---|
| w/o Router | 5.3 | 27.2 | 68.5/80.2 | 48.9/78.7 | 67.4/79.4 | 72.9/93.1 | 60.0/79.8 |
| MLP | 5.3 | 27.2 | 70.7/81.8 | 53.2/80.4 | 71.2/83.1 | 75.2/94.3 | 63.3/81.6 |
| MoGE | 5.3 | 27.2 | 73.4/85.2 | 59.6/85.6 | 70.0/82.8 | 74.7/94.0 | 67.1/85.4 |

Table 7. Ablations for Full Fine-tuning(FF). "R" denotes Random Initialization. "P" denotes pre-trained weights from RECON.

| Method | Car | Pedestrain | Van | Cyclist | Mean |
|---|---|---|---|---|---|
| FF-R | 70.6/82.6 | 51.3/80.6 | 64.1/76.6 | 65.4/92.3 | 61.5/81.4 |
| FF-P | 69.8/83.6 | 53.6/81.6 | 62.2/73.6 | 65.8/90.8 | 62.0/82.0 |
| w/o FF | 73.4/85.2 | 59.6/85.6 | 70.0/82.8 | 74.7/94.0 | 67.1/85.4 |

Table 8. Ablations for Pretrained Model.

| Methods | Car | Pedestrian | Van | Cyclist | Mean |
|---|---|---|---|---|---|
| TrackAny3D-PointMAE | 73.3/85.0 | 59.1/85.4 | 69.7/81.3 | 71.8/92.1 | 66.8/85.0 |
| TrackAny3D-RECON | 73.4/85.2 | 59.6/85.6 | 70.0/82.8 | 74.7/94.0 | 67.1/85.4 |

## References

[1] 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. 1

[2] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 1

[3] Jiahao Nie, Zhiwei He, Xudong Lv, Xueyi Zhou, Dong-Kyu Chae, and Fei Xie. Towards category unification of 3d single object tracking on point clouds. *arXiv preprint arXiv:2401.11204*, 2024. 1

[4] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 1

[5] Tian-Xing Xu, Yuan-Chen Guo, Yu-Kun Lai, and Song-Hai Zhang. Mbptrack: Improving 3d point cloud tracking with memory networks and box priors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9911–9920, 2023. 1