SimpleTrack: Understanding and Rethinking 3D Multi-object Tracking

Ziqi Pang[1(✉)], Zhichao Li[2], and Naiyan Wang[2]

[1] UIUC, Urbana-Champaign, USA
ziqip2@illinois.edu
[2] TuSimple, California, USA

Abstract. 3D multi-object tracking (MOT) has witnessed numerous novel benchmarks and approaches in recent years, especially those under the "tracking-by-detection" paradigm. Despite their progress and usefulness, an in-depth analysis of their strengths and weaknesses is not yet available. In this paper, we summarize current 3D MOT methods into a unified framework by decomposing them into four constituent parts: pre-processing of detection, association, motion model, and life cycle management. We then ascribe the failure cases of existing algorithms to each component and investigate them in detail. Based on the analyses, we propose corresponding improvements which lead to a strong yet simple baseline: SimpleTrack. Comprehensive experimental results on Waymo Open Dataset and nuScenes demonstrate that our final method could achieve new state-of-the-art results with minor modifications. Furthermore, we take additional steps and rethink whether current benchmarks authentically reflect the ability of algorithms for real-world challenges. We delve into the details of existing benchmarks and find some intriguing facts. Finally, we analyze the distribution and causes of remaining failures in SimpleTrack and propose future directions for 3D MOT. Our code is at https://github.com/tusen-ai/SimpleTrack.

Keywords: 3D multi-object tracking · Autonomous driving

1 Introduction

Multi-object tracking (MOT) is a composite task in computer vision, combining both the aspects of localization and identification. Given its complex nature, MOT systems generally involve numerous interconnected parts, such as the selection of detections, the data association, the modeling of object motions, etc. Each of these modules has its special treatment and can significantly affect the system performance as a whole. Therefore, we would like to ask *which components in 3D MOT play the most important roles, and how can we improve them?*

Z. Pang—This work is complete during the first author's internship at TuSimple.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-25056-9_43.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
L. Karlinsky et al. (Eds.): ECCV 2022 Workshops, LNCS 13801, pp. 680–696, 2023.
https://doi.org/10.1007/978-3-031-25056-9_43

Bearing such objectives, we revisit the current 3D MOT algorithms [3,10,13,14,30,32,40,46,47]. These methods mostly adopt the "tracking by detection" paradigm, where they directly take the bounding boxes from 3D detectors and build up tracklets across frames. We first break them down into four individual modules and examine each of them: pre-processing of input detections, motion model, association, and life cycle management. Based on this modular framework, we locate and ascribe the failure cases of 3D MOT to the corresponding components and discover several overlooked issues in the previous designs.

First, we find that inaccurate input detections may contaminate the association. However, purely pruning them by a score threshold will sacrifice the recall. Second, we find that the similarity metric defined between two 3D bounding boxes need to be carefully designed. Neither distance-based nor simple IoU works well. Third, the object motion in 3D space is more predictable than that in the 2D image space. Therefore, the consensus between motion model predictions and even poor observations (low score detections) could well indicate the existence of objects. Illuminated by these observations, we propose several simple yet non-trivial solutions. The evaluation on Waymo Open Dataset [37] and nuScenes [8] suggests that our final method "SimpleTrack" is competitive among the 3D MOT algorithms (in Table 4 and Table 5).

Besides analyzing 3D MOT algorithms, we also reflect on current benchmarks. We emphasize the need for high-frequency detections and the proper handling of output tracklets in evaluation. To better understand the upper bound of our method, we further break down the remaining errors based on ID switch and MOTA metrics. We believe these observations could inspire the better design of algorithms and benchmarks.

In brief, our contributions are as follow:

- We analyze each component in 3D MOT methods and their failure cases, based on a decomposition of "tracking-by-detection" 3D MOT framework.
- We propose corresponding treatments for each module and combine them into a simple baseline. The results are competitive on the Waymo Open Dataset and nuScenes.
- We also analyze existing 3D MOT benchmarks and explain the potential influences of their designs. We hope that our analyses could shed light for future research.

2 Related Work

Most 3D MOT methods [3,10,13,14,30,32,40,46,47] adopt the "tracking-by-detection" framework because of the strong power of detectors. We first summarize the representative 3D MOT work and then highlight the connections and distinctions between 3D and 2D MOT.

2.1 3D MOT

Many 3D MOT methods are composed of rule-based components. AB3DMOT [40] is the common baseline of using IoU for association and a Kalman filter as the motion model. Its notable followers mainly improve on the association part: Chiu *et al.* [10] and CenterPoint [46] replace IoU with Mahalanobis and L2 distance, which performs better on nuScenes [8]. Some others notice the importance of life cycle management, where

CBMOT [3] proposes a score-based method to replace the "count-based" mechanism, and Pöschmann *et al.* [30] treats 3D MOT as optimization problems on factor graphs. Despite the effectiveness of these improvements, a systematic study on 3D MOT methods is in great need, especially where these designs suffer and how to make further improvements. To this end, our paper seeks to meet the expectations.

Different from the methods mentioned above, many others attempt to solve 3D MOT with fewer manual designs. [2,9,18,41] leverage rich features from RGB images for association and life cycle control, [32] manages to fuse the object proposals from different modalities, and Chiu *et al.* [9] specially uses neural networks to handle the feature fusion, association metrics, and tracklet initialization. Recently, OGR3MOT [47] follows Guillem *et al.* [7] and solves 3D MOT with graph neural networks (GNN) in an end-to-end manner, focusing on the data association and the classification of active tracklets, especially. SDVTracker [13] systematically investigates 3D MOT, and proposes to extract descriptors of agents for association and update the tracks in an interaction-aware manner. Compared to SDVTracker, which is innovative analysis on 3D MOT, SimpleTrack focuses on the priors of 3D MOT and rule-based 3D MOT systems, and our analyses are based on public datasets Waymo Open Dataset and nuScenes.

2.2 2D MOT

2D MOT shares the common goal of data association with 3D MOT. Some notable attempts include probabilistic approaches [1,19,33,35], dynamic programming [12], bipartite matching [6], min-cost flow [4,49], convex optimization [29,38,39,48], and conditional random fields [45]. With the rapid progress of deep learning, many methods [7,15–17,22,43] learn the matching mechanisms and others [20,23,24,26,28] learn the association metrics.

Similar to 3D MOT, many 2D trackers [5,25,36,51] also benefit from the enhanced detection quality and adopt the "tracking-by-detection" paradigm. However, the objects on RGB images have varied sizes because of scale variation; thus, they are harder for association and motion models. But 2D MOT can easily take advantage of rich RGB information and use appearance models [21,22,36,42], which is not available in LiDAR based 3D MOT. In summary, the design of MOT methods should fit the traits of each modality.

3 3D MOT Pipeline

In this section, we decompose 3D MOT methods into the following four parts. An illustration is in Fig. 1.

Pre-processing of Input Detections. It pre-processes the bounding boxes from detectors and selects the ones to be used for tracking. Some exemplar operations include selecting the bounding boxes with scores higher than a certain threshold. (In "Pre-processing" Fig. 1, some redundant bounding boxes are removed.)

Motion Model. It predicts and updates the states of objects. Most 3D MOT methods [3,10,40] directly use the Kalman filter, and CenterPoint [46] uses the velocities predicted by detectors from multi-frame data. (In "Prediction" and "Motion Model Update" Fig. 1.)

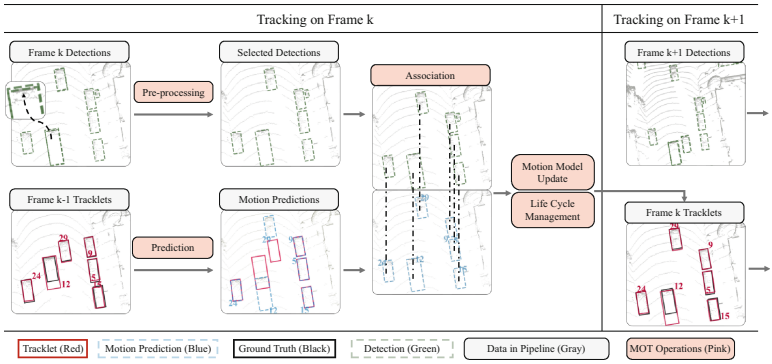

Fig. 1. 3D MOT pipeline. For simplicity, we only visualize the steps between frame k and frame k+1. Best view in color.

Association. It associates the detections with tracklets. The association module involves two steps: similarity computation and matching. The similarity measures the distance between a pair of detection and tracklet, while the matching step solves the correspondences based on the pre-computed similarities. AB3DMOT [40] proposes the baseline of using IoU with Hungarian algorithm, while Chiu *et al.* [10] uses Mahalanobis distance and greedy algorithm, and CenterPoint [46] adopts the L2 distance. (In "Association" Fig. 1.)

Life Cycle Management. It controls the "birth", "death" and "output" policies. "Birth" determines whether a detection bounding box will be initialized as a new tracklet; "Death" removes a tracklet when it is believed to have moved out of the attention area; "Output" decides whether a tracklet will output its state. Most of the MOT algorithm adopts a simple count-based rule [10,40,46], and CBMOT [3] improves birth and death by amending the logic of tracklet confidences. (In "Life Cycle Management" Fig. 1.)

4 Analyzing and Improving 3D MOT

In this section, we analyze and improve each module in the 3D MOT pipeline. For better clarification, we ablate the effects of every modification by removing it from the final variant of SimpleTrack. By default, the ablations are all on the validation split with the CenterPoint [46] detection. We also provide additive ablation analyses and the comparison with other methods in Sect. 4.5.

4.1 Pre-processing

To fulfill the recall requirements for detection AP, current detectors usually output a large number of bounding boxes with scores roughly indicating their quality. However, if these boxes are treated equally in the association step of 3D MOT, the bounding boxes with low quality or severe overlapping, which is not fully addressed by a loose NMS in 3D detectors, may deviate the trackers to select the inaccurate detection for extending or

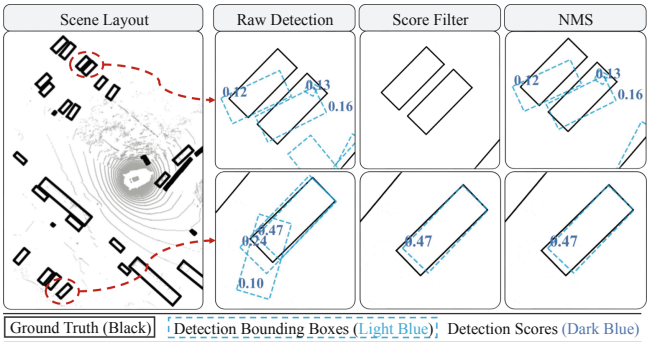

Fig. 2. Comparison between score filtering and NMS. To remove the redundant bounding boxes on row 2, score filtering needs at least a 0.24 threshold, but this will eliminate the detections on row 1. However, NMS can well satisfy both by removing the overlapping on row 2 and maintaining the recall on row 1.

Table 1. Left: ablation for NMS on nuScenes. Right: ablation for NMS on WOD.

NMS	AMOTA↑	AMOTP↓	MOTA↑	IDS ↓
×	0.673	0.574	0.581	557
✓	**0.687**	**0.573**	**0.592**	**519**

NMS	Vehicle			Pedestrian		
	MOTA↑	MOTP↓	IDS(%)↓	MOTA↑	MOTP↓	IDS(%)↓
×	0.5609	**0.1681**	0.09	0.4962	**0.3090**	5.00
✓	**0.5612**	**0.1681**	**0.08**	**0.5776**	0.3125	**0.42**

forming tracklets (as in the "raw detection" of Fig. 2). Such a gap between the detection and MOT task needs careful treatment.

3D MOT methods commonly use confidence scores to filter out the low-quality detections and improve the precision of input bounding boxes. However, such an approach may be detrimental to the recall as they directly abandon the objects with poor observations (top row in Fig. 2). It is also especially harmful to metrics like AMOTA, which needs the tracker to use low score bounding boxes to fulfill the recall requirements.

To improve the precision without significantly decreasing the recall, our solution is simple and direct: we apply ***stricter*** non-maximum suppression (NMS) to the input detections. Please note that 3D detectors [46] already applies NMS prior to 3D MOT, however, we emphasize that the IoU threshold for NMS should be higher in 3D MOT, compared to detection. As shown in the right of Fig. 2, the NMS operation alone can effectively eliminate the overlapped low-quality bounding boxes while keeping the diverse low-quality observations, even on regions like sparse points or occlusion. *Therefore, by adding NMS to the pre-processing module, we could roughly keep the recall, but greatly improves the precision and benefits MOT.*

On WOD, our stricter NMS operation removes 51% and 52% bounding boxes for vehicles and pedestrians and nearly doubles the precision: 10.8% to 21.1% for vehicles, 5.1% to 9.9% for pedestrians. At the same time, the recall drops relatively little from 78% to 74% for vehicles and 83% to 79% for pedestrians. According to Table 1, this largely benefits the performance, especially on the pedestrian (right part of Table 1), where the object detection task is harder.

Table 2. Left: comparison of motion models on Waymo Open Dataset. "KF" denotes Kalman filters; "CV" denotes constant velocity model; "KF-PD" denotes the variant using Kalman filter only for motion prediction. Right: comparison of motion models on nuScenes. Details in Sect. 4.2.

Method	Vehicle			Pedestrian		
	MOTA↑	MOTP↓	IDS(%)↓	MOTA↑	MOTP↓	IDS(%)↓
KF	**0.5612**	**0.1681**	**0.08**	**0.5776**	**0.3125**	**0.42**
CV	0.5515	0.1691	0.14	0.5661	0.3159	0.58
KF PD	0.5516	0.1691	0.14	0.5654	0.3158	0.63

Method	AMOTA↑	AMOTP↓	MOTA↑	IDS ↓
KF	0.687	0.573	**0.592**	519
CV	**0.690**	**0.564**	**0.592**	**516**

4.2 Motion Model

Motion models depict the motion status of tracklets. They are mainly used to predict the candidate states of objects in the next frame, which are the proposals for the following association step. Furthermore, the motion models like the Kalman filter can also potentially refine the states of objects. In general, there are two commonly adopted motion models for 3D MOT: Kalman filter (KF), *e.g.* AB3DMOT [40], and constant velocity model (CV) with predicted speeds from detectors, *e.g.* CenterPoint [46]. The advantage of KF is that it could utilize the information from multiple frames and provide smoother results when facing low-quality detection. Meanwhile, CV deals better with abrupt and unpredictable motions with its explicit speed predictions, but its effectiveness on motion smoothing is limited. In Table 2, we compare the two of them on WOD and nuScenes, which provides clear evidence for our claims.

In general, these two motion models demonstrate similar performance. On nuScenes, CV marginally outperforms KF, while it is the opposite on WOD. The advantages of KF on WOD mainly come from the refinement for the bounding boxes. To verify this, we implement the "KF-PD" variant, which uses KF only for providing motion predictions prior to association, and the outputs are all original detections. Eventually, the marginal gap between "CV" and "KF-PD" in Table 2 left supports our claim. On nuScenes, the CV motion model is slightly better due to the lower frame rates on nuScenes (2 Hz). To prove our conjecture, we apply KF and CV both under a higher 10 Hz setting on nuScenes[1], and KF marginally outperforms CV by 0.696 versus 0.693 in AMOTA.

To summarize, *the Kalman Filter fits better for high-frequency cases because of more predictable motions, and the constant velocity model is more robust for low-frequency scenarios with explicit speed prediction.* Since inferring velocities is not yet common for detectors, we adopt the Kalman filter for without loss of generality.

4.3 Association

Association Metrics: 3D GIoU IoU based [40] and distance based [10,46] association metrics are the two prevalent choices in 3D MOT. As in Fig. 3, they have typical but different failure modes. IoU computes the overlapping ratios between bounding boxes, so it cannot connect the detections and motion predictions if the IoU between them are all zeros, which are common at the beginnings of tracklets or on objects with abrupt

[1] Please check Sect. 5.1 for how we 10 Hz settings on nuScenes.

motions (the left of Fig. 3). The representatives for distance-based metrics are Mahalanobis [10] and L2 [46] distances. With larger distance thresholds, they can handle the failure cases of IoU based metrics, but they may not be sensitive enough for nearby detection with low quality. We explain such scenarios on the right of Fig. 3. On frame k, the blue motion prediction has smaller L2 distances to the green false positive detection, thus it is wrongly associated. Illuminating by such example, we conclude that the distance-based metrics lack discrimination on orientations, which is just the advantage of IOU based metrics.

To get the best of two worlds, we propose to generalize "Generalized IoU" (GIoU) [34] to 3D for association. Briefly speaking, for any pair of 3D bounding boxes B_1, B_2, their 3D GIoU is as Eq. 1, where I, U are the intersection and union of B_1 and B_2. Furthermore, the convex hull of B_1, B_2 is denoted by C (short for *convex hull*). V represents the volume of a polygon. We set GIoU > -0.5 as the threshold for every category of objects on both WOD and nuScenes for this pair of associations to enter the subsequent matching step.

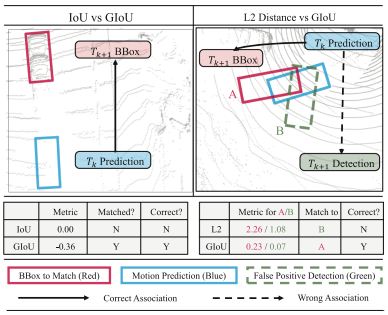

Fig. 3. Illustration of association metrics. IoU (left)/L2 Distance (right) versus GIoU. Details are in Sect. 4.3. (Color figure online)

$$
\begin{aligned}
V_U &= V_{B_1} + V_{B_2} - V_I, \\
\mathbf{GIoU}(B_1, B_2) &= V_I / V_U - (V_C - V_U)/V_C.
\end{aligned}
\tag{1}
$$

As in Fig. 3, the GIoU metric can handle both patterns of failures. The quantitative results in Fig. 4 also show the ability of GIoU for improving the association on both WOD and nuScenes.

Matching Strategies. Generally speaking, there are two approaches for the matching between detections and tracklets: 1) Formulating the problem as a bipartite matching

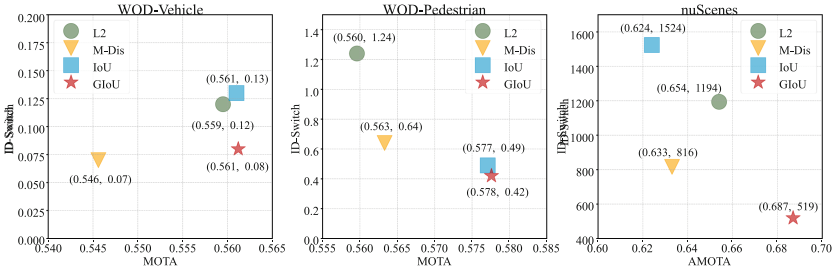

Fig. 4. Comparison of association metrics on WOD (left & middle) and nuScenes (right). "M-Dis" is the short for Mahalanobis distance. The best method is closest to the bottom-right corner, having the lowest ID-Switches and highest MOTA/AMOTA. IoU and GIoU use Hungarian algorithm for matching, while L2/M-Dis use greedy algorithm (explained in Sect. 4.3).

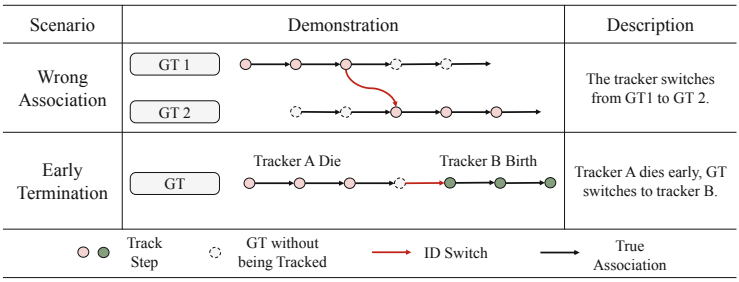

Fig. 6. Illustration for two major types of ID-Switches.

problem, and then solving it using Hungarian algorithm [40]. 2) Iteratively associating the nearest pairs by greedy algorithm [10,46].

We find that these two methods heavily couples with the association metrics: IoU based metrics are fine with both, while distance-based metrics prefer greedy algorithms. We hypothesize that the reason is that the range of distance-based metrics are large, thus methods optimizing global optimal solution, like the Hungarian algorithm, may be adversely affected by outliers. In Fig. 5, we experiment with all the combinations between matching strategies and association metrics on WOD. As demonstrated, IoU and GIoU function well for both strategies, while Mahalanobis and L2 distance demand greedy algorithm, which is also consistent with the conclusions from previous work [10].

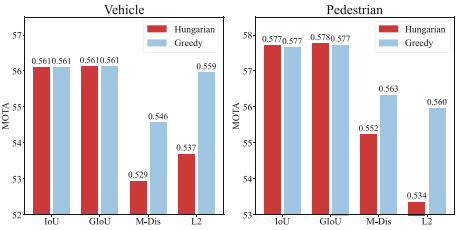

Fig. 5. Comparison of matching strategies on WOD.

4.4 Life Cycle Management

We analyze all the ID-Switches on WOD[2], and categorize them into two groups as in Fig. 6: wrong association and early termination. Different from the major focus of many work, which is association, we find that the early termination is actually the dominating cause of ID-Switches: 95% for vehicle and 91% for pedestrian. Among the early terminations, many of them are caused by point cloud sparsity and spatial occlusion. To alleviate this issue, we utilize the free yet effective information: consensus between motion models and detections with low scores. *These bounding boxes are usually of low localization quality, however they are strong indication of the existence of objects if they agree with the motion predictions.* Then we use these to extend the lives of tracklets.

Bearing such motivation, we propose "Two-stage Association." Specifically, we apply two rounds of association with different score thresholds: a low one $\mathbf{T}_l$ and a

[2] We use py-motmetrics [11] for the analysis.

Detection with Scores		0.8	0.6	0.4	0.2	0.8
Frame Number		1	2	3	4	5
One-stage	Action	Initialize	Match	Predict	Death	Initialize
	Object ID	A	A	A	A	B
	ID-Switch	0	0	0	0	1
Two-stage	Action	Initialize	Match	Extend	Extend	Match
	Object ID	A	A	A	A	A
	ID-Switch	0	0	0	0	0

Fig. 7. Comparison for "One-stage" and "Two-stage" association with a hypothetical example. "Extend" means "extending the life cycles," and "Predict" means "using motion predictions due to no association." Suppose $\mathbf{T}_h = 0.5$ and $\mathbf{T}_l = 0.1$ are the score thresholds, the "one-stage" method early terminates the tracklet because of consecutively lacking associations. Details in Sect. 4.4.

high one $\mathbf{T}_h$ (*e.g.* 0.1 and 0.5 for pedestrian on WOD). In stage one, we use the identical procedure as most current algorithms [10,40,46]: only the bounding boxes with scores higher than $\mathbf{T}_h$ are used for association. In stage two, we focus on the tracklets unmatched to detections in stage one and relax the conditions on their matches: detections having scores larger than $\mathbf{T}_l$ will be sufficient for a match. If the tracklet is successfully associated with one bounding box in stage two, it will still keep being alive. However, as the low score detections are generally in poor quality, we don't output them to avoid false positives, and they are also not used for updating motion models. Instead, we use motion predictions as the latest tracklet states, replacing the low quality detections.

We intuitively explain the differences between our "Two-stage Association" and traditional "One-stage Association" in Fig. 7. Suppose $\mathbf{T} = 0.5$ is the original score threshold for filtering detection bounding boxes, the trackers will then neglect the boxes with scores 0.4 and 0.2 on frames 3 and 4, which will die because of lacking matches in continuous frames and this eventually causes the final ID-Switch. In comparison, our two-stage association can maintain the active state of the tracklet.

Table 3. Ablation for "Two-stage Association" on WOD. "One" and "Two" denotes the previous one-stage association and our two-stage association methods. Details in Sect. 4.4.

Method	Vehicle			Pedestrian		
	MOTA↑	MOTP↓	IDS(%)↓	MOTA↑	MOTP↓	IDS(%)↓
One	0.5567	0.1682	0.46	0.5718	**0.3125**	0.96
Two	**0.5612**	**0.1681**	**0.08**	**0.5776**	**0.3125**	**0.42**

In Table 3, our approach greatly decreases the ID-Switches without hurting the MOTA. This proves that SimpleTrack is effective in extending the life cycles by using detections more flexibly. Parallel to our work, a similar approach is also proven to be useful for 2D MOT [50].

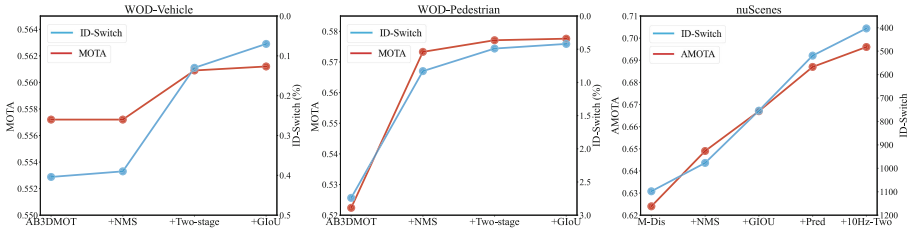

Fig. 8. Improvements from SimpleTrack on WOD (left & middle) and nuScenes (right). We use the common baselines of AB3DMOT [40] on WOD and Chiu *et al.* [10] on nuScenes. For nuScenes, the improvements of "10 Hz-Two" (10 Hz detection and two-stage association) is in Sect. 5.1, and "Pred" (outputting motion model predictions) is in Sect. 5.2. The names for modifications are on the x-axis. Better MOTA and ID-Switch values are higher on the y-axis for clearer visualization.

Table 4. Comparison on WOD test split (L2). CenterPoint [46] detections are used. We list the methods using public detection. For AB3DMOT [40] and Chiu *et al.* [10], we report their best leaderboard entries.

Method	Vehicle			Pedestrian		
	MOTA↑	MOTP↓	IDS(%)↓	MOTA↑	MOTP↓	IDS(%)↓
AB3DMOT [40]	0.5773	**0.1614**	0.26	0.5380	0.3163	0.73
Chiu *et al.* [10]	0.4932	0.1689	0.62	0.4438	0.3227	1.83
CenterPoint [46]	0.5938	0.1637	0.32	0.5664	0.3116	1.07
SimpleTrack	**0.6030**	0.1623	**0.08**	**0.6013**	**0.3114**	**0.40**

4.5 Integration of SimpleTrack

In this section, we integrate the techniques mentioned above into the unified SimpleTrack and demonstrate how they improve the performance step by step.

In Fig. 8, we illustrate how the performance of 3D MOT trackers improve from the baselines. On WOD, although the properties of vehicles and pedestrian are much different, each technique is applicable to both. On nuScenes, every proposed improvement is also effective for both the AMOTA and ID-Switch.

We also report the test set performance and compare with other 3D MOT methods. SimpleTrack could achieve new state-of-the-art results with nominal cost, running at 120 FPS after detectors. (in Table 4, Table 5).[3]

5 Rethinking NuScenes

Besides the techniques mentioned above, we delve into the design of benchmarks. The benchmarks greatly facilitate the development of research and guide the designs of algorithms. Contrasting WOD and nuScenes, we find that despite more than 70% of vehicles

[3] Validation split comparisons are in the supplementary.

Table 5. Comparison on nuScenes test split. CenterPoint [46] detections are used. We list the methods using public detection. For CBMOT [3] and OGR3MOT [47], we report their numbers with CenterPoint [46] detection. Our numbers using 2 Hz and 10 Hz frame rate detections are reported (details of 10 Hz setting are in Sect. 5).

Methods	AMOTA↑	AMOTP↓	MOTA↑	IDS ↓
AB3DMOT [40]	0.151	1.501	0.154	9027
Chiu *et al.* [10]	0.550	0.798	0.459	776
CenterPoint [46]	0.638	0.555	0.537	760
CBMOT [3]	0.649	0.592	0.545	557
OGR3MOT citech43graphmot	0.656	0.620	0.554	**288**
SimpleTrack (2 Hz)	0.658	0.568	0.557	609
SimpleTrack (10 Hz)	**0.668**	**0.550**	**0.566**	575

staying static all the time, which may lead to biases to the evaluation, WOD is closer to real-world scenarios. As for nuScenes, two aspects are critical: 1) The frame rate of nuScenes 2 Hz, while WOD 10 Hz. Such low frequency adds unnecessary difficulties to 3D MOT (Sect. 5.1). 2) The evaluation of nuScenes requires high recalls with low score thresholds. And it also pre-processes the tracklets with interpolation, which encourages trackers to output the confidence scores reflecting the entire tracklet quality, but not the frame quality (Sect. 5.2).

We hope these two findings could inspire the community to rethink the benchmarks and evaluation protocols of 3D tracking. *And such modifications may help: a. support 10 Hz setting; b. discard the "interpolation" in evaluation API.*

Table 6. Frequency comparison of benchmarks.

Benchmark	Data	Annotation	Model
Waymo Open Dataset	10 Hz	10 Hz	10 Hz
nuScenes	20 Hz	2 Hz	2 Hz

5.1 Detection Frequencies

Tracking generally benefits from higher frame rates, because motion is more predictable in short intervals.

We compare the frequencies of point clouds, annotations, and common MOT frame rates on the two benchmarks in Table 6. On nuScenes, it 20 Hz point clouds but 2 Hz annotations. This leads to most common detectors and 3D MOT algorithms work 2 Hz, even they actually utilize all 20 Hz LiDAR data and operate faster 2 Hz. Therefore, we investigate the effect of high-frequency data as follows. Although the information is more abundant with high frequency (HF) frames, it is non-trivial to incorporate them because nuScenes only evaluates on the low-frequency frames, which we refer to as

"evaluation frames." In Table 7, simply using all 10 Hz frames does not improve the performance. This is because the low-quality detection on the HF frames may deviate the trackers and hurt the performance on the sampled evaluation frames. To overcome this issue, we explore by first applying the "One-stage Association" on HF frames, where only the bounding boxes with scores larger than $\mathbf{T}_h = 0.5$ are considered and used for motion model updating. We then adopt the "Two-stage Association" (described in Sect. 4.4) by using the boxes with scores larger than $\mathbf{T}_l = 0.1$ to extend the tracklets. As in Table 7, our approach significantly improves both the AMOTA and ID-Switches. We also try to even increase the frame rate 20 Hz, but this barely leads to further improvements due to the deviation issue. So SimpleTrack uses 10 Hz setting in our final submission to the test set.[4]

5.2 Tracklet Interpolation

We start with the evaluation protocol on nuScenes, where they interpolate the input tracklets to fill in the missing frames and change all the scores with their tracklet-average scores as illustrated in Fig. 9. Such interpolation on nuScenes encourages the trackers to treat tracklet quality holistically and output calibrated quality-aware scores. However the quality of boxes may vary a lot across frames even for the same tracklet, thus we suggest depicting the quality of a tracklet by only one score is imperfect. Moreover, future information is also introduced in this interpolation step and it changes the tracklet results. This could also bring the concern on whether the evaluation setting is still a fully online one which is crucial for autonomous driving.

To prove this argument, we output the motion model predictions for frames and tracklets without associated detection bounding boxes, and empirically assign them lower scores than any other detection. In our case, their scores are $0.01 \times S_P$, where S_P is the confidence score of the tracklet in the previous frame. As shown in Fig. 9, Our approach can explicitly penalize the low-quality tracklets, which generally contain more missing boxes replaced by motion model predictions. In Table 8, this simple experiment improves the overall recall and AMOTA. Moreover, some attempts [3,47] have changed tracklet scores without explicitly declaring the "interpolation" issue which should be aware to ensure the effectiveness of the benchmark.

Table 7. MOT with higher frame rates on nuScenes. "10 Hz" is the vanilla baseline of using all the detections on high frequency (HF) frames. "-One" denotes "One-stage," and "-Two" denotes "Two-stage." Details in Sect. 5.1.

Setting	AMOTA↑	AMOTP↓	MOTA↑	IDS ↓
2 Hz	0.687	0.573	0.592	519
10 Hz	0.687	0.548	0.599	512
10 Hz - One	**0.696**	0.564	**0.603**	450
10 Hz - Two	**0.696**	**0.547**	0.602	**403**
20 Hz - Two	0.690	**0.547**	0.598	416

[4] Because of the submission time limits to nuScenes test set, we are only able to report the "10 Hz-One" variant in Table 5. It will be updated to "10 Hz-Two" once we had the chance.

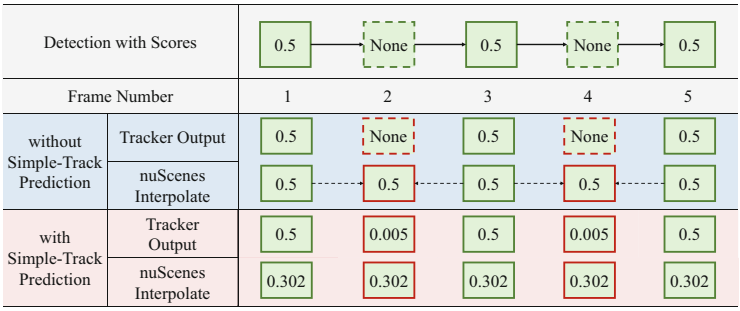

Fig. 9. How the motion predictions and nuScenes interpolation changes tracklet scores. Dashed arrows are the directions for interpolation. On Frame 2 and 4 the boxes with score 0.05 are our motion predictions. The "0.5" and "0.302" are the tracklet-average scores with or without motion predictions. Details in Sect. 5.2.

Table 8. Improvement from "outputting motion model predictions" on nuScenes (2 Hz detections for ablation).

Predictions	AMOTA↑	AMOTP↓	MOTA↑	IDS ↓	RECALL↑
×	0.667	0.612	0.572	754	0.696
✓	**0.687**	**0.573**	**0.592**	**519**	**0.725**

6 Error Analyses

In this section, we conduct analyses on the remaining failure cases of SimpleTrack and propose potential future directions for improving "tracking by detection" paradigm. Without loss of generality, we use WOD as an example.

6.1 Upper Bound Experiment Settings

To quantitatively evaluate the causes of failure cases, we contrast SimpleTrack with two different oracle variants. The results are summarized in Table 9.

GT Output erases the errors caused by "output" policy. We compute the IoU between the bounding boxes from SimpleTrack with the GT boxes at the "output" stage, then use the IoU to decide if a box should be output instead of the detection score. [5]

GT All is the upper bound of tracking performance with CenterPoint boxes. We greedily match the detections from CenterPoint to GT boxes, keep the true positive and assign them ground-truth ID.

[5] The ID-Switch increases because we output more bounding boxes and IDs. The 0.003 false positives in pedestrians are caused by boxes matched with the same GT box in crowded scenes.

6.2 Analyses for "Tracking by Detection"

ID-Switches. We break down the causes of ID-Switches as in Fig. 6. Although early termination has been greatly decreased by the scale of 86% for vehicle and 70% for pedestrian with "Two-stage Association," it still takes up 88% and 72% failure cases in the remaining ID-Switches in SimpleTrack for vehicle and pedestrian, respectively. We inspect these cases and discover that most of them result from long-term occlusion or the returning of objects from being temporarily out of sight. Therefore, in addition to improving the association, potential future work can develop appearance models like in 2D MOT [21,22,36,42] or silently maintain their states to re-identify these objects after they are back.

Table 9. Oracle experiments on WOD.

Method	Vehicle				Pedestrian			
	MOTA↑	IDS(%)↓	FP↓	FN↓	MOTA↑	IDS(%)↓	FP↓	FN↓
SimpleTrack	0.561	0.078	0.104	0.334	0.578	0.425	0.109	0.309
GT Output	0.741	0.104	**0.000**	0.258	0.778	0.504	0.003	0.214
GT All	**0.785**	**0.000**	**0.000**	**0.215**	**0.829**	**0.000**	**0.000**	**0.171**

FP and FN. The "GT All" in Table 9 shows the upper bound for MOT with CenterPoint [46] detection, and we analyze the class of vehicle for example. Even with "GT All" the false negatives are still 0.215, which are the detection FN and can hardly be fixed under the "tracking by detection" framework. Comparing "GT All" and SimpleTrack, we find that the tracking algorithm itself introduces 0.119 false negatives. We further break them down as follows. Specifically, the difference between "GT Output" and "GT ALL" indicates that the 0.043 false negatives are caused by the uninitialized tracklets resulting from NMS and score threshold in pre-processing. The others come from life-cycle management. The "Initialization" requires two frames of accumulation before outputting a tracklet, which is same as AB3DMOT [40]. This yields a marginal 0.005 false negatives. Our "Output" logic uses detection score to decide output or not, taking up the false negatives number 0.076. Based on these analyses, we can conclude that the gap is mainly caused by the inconsistency between the scores and detection quality. By using historical information, 3D MOT can potentially provide better scores compared to single frame detectors, and this has already drawn some recent attention [3,47].

7 Conclusions and Future Work

In this paper, we decouple the "tracking by detection" 3D MOT algorithms into several components and analyze their typical failures. With such insights, we propose corresponding enhancements of using *NMS*, *GIoU*, and *Two-stage Association*, which lead to our SimpleTrack. In addition, we also rethink the frame rates and interpolation preprocessing in nuScenes. We eventually point out several possible future directions for "tracking by detection" 3D MOT.

However, beyond the "tracking by detection" paradigm, there are also branches of great potential. For better bounding box qualities, 3D MOT can refine them using long term information [27,31,44], which are proven to outperform the detections based only on local frames. The future work can also transfer the current manual rule-based methods into learning-based counterparts, *e.g.* using learning based intra-frame mechanisms to replace the NMS, using inter-frame reasoning to replace the 3D GIoU and life cycle management, etc.

References

1. Bar-Shalom, Y., Fortmann, T.E., Cable, P.G.: Tracking and data association (1990)
2. Baser, E., Balasubramanian, V., Bhattacharyya, P., Czarnecki, K.: FANTrack: 3D multi-object tracking with feature association network. In: IV (2019)
3. Benbarka, N., Schröder, J., Zell, A.: Score refinement for confidence-based 3D multi-object tracking. In: IROS (2021)
4. Berclaz, J., Fleuret, F., Turetken, E., Fua, P.: Multiple object tracking using k-shortest paths optimization. IEEE Trans. Pattern Anal. Mach. Intell. **33**(9), 1806–1819 (2011)
5. Bergmann, P., Meinhardt, T., Leal-Taixe, L.: Tracking without bells and whistles. In: ICCV (2019)
6. Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple online and realtime tracking. In: ICIP (2016)
7. Braso, G., Leal-Taixe, L.: Learning a neural solver for multiple object tracking. In: CVPR (2020)
8. Caesar, H., et al.: nuScenes: a multimodal dataset for autonomous driving. In: CVPR (2020)
9. Chiu, H., Li, J., Ambrus, R., Bohg, J.: Probabilistic 3D multi-modal, multi-object tracking for autonomous driving. In: ICRA (2021)
10. Chiu, H.k., Prioletti, A., Li, J., Bohg, J.: Probabilistic 3D multi-object tracking for autonomous driving. arXiv:2001.05673 (2020)
11. py-motmetrics Contributors: py-motmetrics. https://github.com/cheind/py-motmetrics
12. Fleuret, F., Berclaz, J., Lengagne, R., Fua, P.: Multicamera people tracking with a probabilistic occupancy map. IEEE Trans. Pattern Anal. Mach. Intell. **30**(2), 267–282 (2007)
13. Gautam, S., Meyer, G.P., Vallespi-Gonzalez, C., Becker, B.C.: Sdvtracker: real-time multi-sensor association and tracking for self-driving vehicles. arXiv preprint arXiv:2003.04447 (2020)
14. Genovese, A.F.: The interacting multiple model algorithm for accurate state estimation of maneuvering targets. J. Hopkins APL Tech. Dig. **22**(4), 614–623 (2001)
15. He, J., Huang, Z., Wang, N., Zhang, Z.: Learnable graph matching: incorporating graph partitioning with deep feature learning for multiple object tracking. In: CVPR (2021)
16. Hornakova, A., Henschel, R., Rosenhahn, B., Swoboda, P.: Lifted disjoint paths with application in multiple object tracking. In: ICML (2020)
17. Jiang, X., Li, P., Li, Y., Zhen, X.: Graph neural ased end-to-end data association framework for online multiple-object tracking. arXiv preprint arXiv:1907.05315 (2019)
18. Kim, A., Osep, A., Leal-Taixé, L.: EagerMOT: 3D multi-object tracking via sensor fusion. arxiv:2104.14682 (2021)
19. Kim, C., Li, F., Ciptadi, A., Rehg, J.M.: Multiple hypothesis tracking revisited. In: ICCV (2015)
20. Lan, L., Tao, D., Gong, C., Guan, N., Luo, Z.: Online multi-object tracking by quadratic pseudo-boolean optimization. In: IJCAI (2016)

21. Leal-Taixé, L., Canton-Ferrer, C., Schindler, K.: Learning by tracking: siamese CNN for robust target association. In: CVPR Workshops (2016)
22. Li, J., Gao, X., Jiang, T.: Graph networks for multiple object tracking. In: WACV (2020)
23. Liang, T., Lan, L., Luo, Z.: Enhancing the association in multi-object tracking via neighbor graph. arXiv preprint arXiv:2007.00265 (2020)
24. Liu, Q., Chu, Q., Liu, B., Yu, N.: GSM: graph similarity model for multi-object trackin. In: IJCAI (2020)
25. Lu, Z., Rathod, V., Votel, R., Huang, J.: RetinaTrack: online single stage joint detection and tracking. In: CVPR (2020)
26. Pang, J., et al.: Quasi-dense similarity learning for multiple object tracking. In: CVPR (2021)
27. Pang, Z., Li, Z., Wang, N.: Model-free vehicle tracking and state estimation in point cloud sequences. In: IROS (2021)
28. Peng, J., et al.: TPM: multiple object tracking with tracklet-plane matching. Pattern Recogn. **107**, 107480 (2020)
29. Pirsiavash, H., Ramanan, D., Fowlkes, C.C.: Globally-optimal greedy algorithms for tracking a variable number of objects. In: CVPR (2011)
30. Pöschmann, J., Pfeifer, T., Protzel, P.: Factor graph based 3D multi-object tracking in point clouds. In: IROS (2020)
31. Qi, C.R., et al.: Offboard 3D object detection from point cloud sequences. In: CVPR (2021)
32. Rangesh, A., Trivedi, M.M.: No blind spots: full-surround multi-object tracking for autonomous vehicles using cameras and lidars. In: IV (2019)
33. Reid, D.: An algorithm for tracking multiple targets. IEEE Trans. Autom. Control **24**(6), 843–854 (1979)
34. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized intersection over union. In: CVPR (2019)
35. Rezatofighi, S.H., Milan, A., Zhang, Z., Shi, Q., Dick, A., Reid, I.: Joint probabilistic data association revisited. In: ICCV (2015)
36. Sadeghian, A., Alahi, A., Savarese, S.: Tracking the untrackable: learning to track multiple cues with long-term dependencies. In: ICCV (2017)
37. Sun, P., et al.: Scalability in perception for autonomous driving: Waymo Open Dataset. arxiv:1912.04838 (2019)
38. Tang, S., Andres, B., Andriluka, M., Schiele, B.: Subgraph decomposition for multi-target tracking. In: CVPR (2015)
39. Tang, S., Andres, B., Andriluka, M., Schiele, B.: Multi-person tracking by multicut and deep matching. In: Hua, G., Jégou, H. (eds.) ECCV 2016. LNCS, vol. 9914, pp. 100–111. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48881-3_8
40. Weng, X., Wang, J., Held, D., Kitani, K.: 3D multi-object tracking: a baseline and new evaluation metrics. In: IROS (2020)
41. Weng, X., Wang, Y., Man, Y., Kitani, K.: GNN3DMOT: graph neural network for 3D multi-object tracking with 2D-3D multi-feature learning. In: CVPR (2020)
42. Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. In: ICIP (2017)
43. Xu, Y., et al.: How to train your deep multi-object tracker. In: CVPR (2020)
44. Yang, B., Bai, M., Liang, M., Zeng, W., Urtasun, R.: Auto4D: learning to label 4D objects from sequential point clouds. arxiv:2101.06586 (2021)
45. Yang, B., Huang, C., Nevatia, R.: Learning affinities and dependencies for multi-target tracking using a CRF model. In: CVPR (2011)
46. Yin, T., Zhou, X., Krähenbühl, P.: Center-based 3D object detection and tracking. In: CVPR (2021)
47. Zaech, J., Dai, D., Liniger, A., Danelljan, M., Gool, L.V.: Learnable online graph representations for 3D multi-object tracking. arXiv:2104.11747 (2021)

48. Roshan Zamir, A., Dehghan, A., Shah, M.: GMCP-tracker: global multi-object tracking using generalized minimum clique graphs. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7573, pp. 343–356. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33709-3_25
49. Zhang, L., Li, Y., Nevatia, R.: Global data association for multi-object tracking using network flows. In: CVPR (2008)
50. Zhang, Y., et al.: ByteTrack: multi-object tracking by associating every detection box. arXiv preprint arXiv:2110.06864 (2021)
51. Zhang, Y., Wang, C., Wang, X., Zeng, W., Liu, W.: FairMOT: on the fairness of detection and re-identification in multiple object tracking. Int. J. Comput. Vis. **129**, 1–19 (2021)