2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

QTrack: Embracing Quality Clues for Robust 3D Multi-Object Tracking

Jinrong Yang[1*], En Yu[1*], Zeming Li[2], Xiaoping Li[1†], and Wenbing Tao[1†]

***Abstract*— 3D Multi-Object Tracking (MOT) has achieved tremendous achievement thanks to the rapid development of 3D object detection and 2D MOT. Recent advanced works generally employ a series of object attributes, *e.g.,* position, size, velocity, and appearance, to provide the clues for the association in 3D MOT. However, these cues may not be reliable due to some visual noise, such as occlusion and blur, leading to tracking performance bottlenecks. To reveal the dilemma, we conduct extensive empirical analysis to expose the key bottleneck of each clue and how they correlate with each other. The analysis results motivate us to efficiently absorb the merits among all cues and adaptively produce an optimal tracking manner. Specifically, we present *Location and Velocity Quality Learning*, which efficiently guides the network to estimate the quality of predicted object attributes. Based on these quality estimations, we propose a quality-aware object association (QOA) strategy to leverage the quality score as an important reference factor for achieving robust association. Despite its simplicity, extensive experiments indicate that the proposed strategy significantly boosts tracking performance by 2.2% AMOTA and our method outperforms all existing state-of-the-art works on nuScenes by a large margin. Moreover, QTrack achieves 51.1%, 54.8% and 56.6% AMOTA tracking performance on the nuScenes test sets with BEVDepth, VideoBEV, and StreamPETR models respectively, which significantly reduces the performance gap between the pure camera and LiDAR-based trackers.**

I. INTRODUCTION

3D Multi-Object Tracking (MOT) has been recently drawing increasing attention since it is widely applied to 3D perception scenes, e.g., autonomous driving, and automatic robots. The 3D MOT task aims at locating objects and associating the targets of the same identities to form tracks. According to the used sensors, existing 3D MOT methods can mainly be categorized into two classes, i.e., camera-based and LiDAR-based schemes. In this paper, we mainly delve into the camera-only scheme since it contains semantic information and is more economical.

Existing 3D MOT methods mostly adopt the tracking-by-detection paradigm. In this regime, a 3d detector is first employed to predict 3D boxes and the corresponding classification scores, and then some post-processing methods (e.g., motion-based [1] or appearance-based) are used to detect targets to form trajectories. In the camera scheme, it is natural to extract objects' discriminative appearance features [2], [3] to represent targets and use the features to measure the similarities among detected targets. However, the procedure of extracting the appearance feature is cumbersome since it requires predicting high-dimensional embedding, which is hard for joint training due to the optimization contradiction between the detection and embedding branches [4]. Moreover, it is difficult to deal with the notorious occlusion and motion blur issues. Some other methods [5], [6] build a motion model (Kalman Filter) to obtain some desired states of tracking clues (e.g, center position, size, size ratio, or rotation) by a linear motion assumption. Nevertheless, this process involves various hyper-parameters (e.g., initialization uncertainty of measurement, state, process, etc.) and executes complex matrix inverse operations. Different from the aforementioned methods, CenterPoint [7] reasonably leverages predicted center locations and velocities of targets for building motion. In detail, it uses a time lag between two moments of observation to multiply the predicted velocity for linear location prediction. Afterwards, the L2 distance among targets acts as a measurement metric for the association procedure. For simplicity, we call this tracking framework CV method. It shows effectiveness to achieve remarkable tracking performance, while only conducting a simple operation (i.e., matrix addition and multiplication) for parallel cost computation.

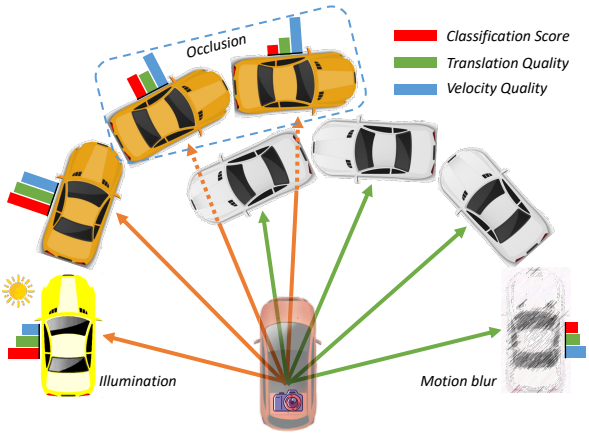
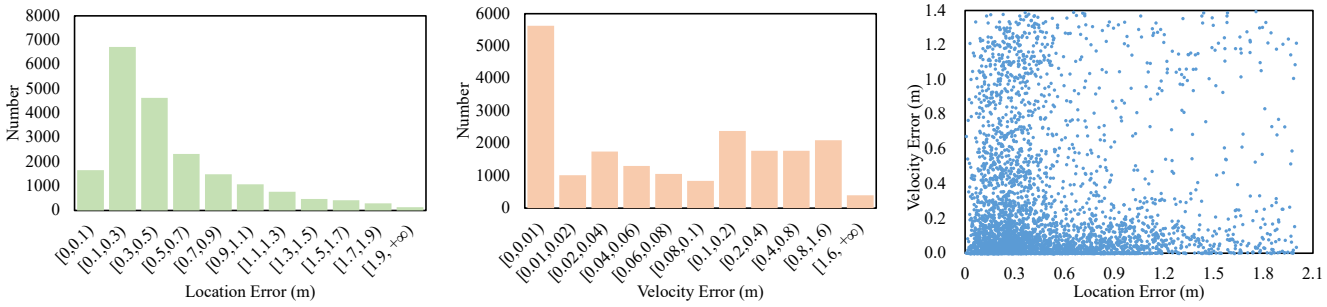


Fig. 1. Illustration of three types of hard cases: (1) illumination of external, (2) occlusion, (3) motion blur. The red, green, and blue pillars are organized to represent the classification score, location quality, and velocity quality, where the higher pillars indicate higher values.

Although the CV framework shows efficiency for 3D MOT task, it relies heavily on the predicted quality of center location and velocity. The requirement may be harsh for the 3D base detector since estimating the center location and velocity of an object from a single image is exactly an ill-posed problem. As shown in Fig. 1, notorious occlusion, motion blur, and the illumination of external issues will significantly

*Equal Contribution ({yangjinrong, yuen}@hust.edu.cn). This work was supported by the CVTE Research and the National Natural Science Foundation of China under Grant 62176096 and Grant 61991412.

[1]Jinrong Yang is with the Huazhong University of Science and Technology and CVTE Research Institute. [1]En Yu, Xiaoping Li, Wenbing Tao are with the Huazhong University of Science and Technology. [2]Zeming Li is with MEGVII Technology. Corresponding author: Xiaoping Li, Wenbing Tao. (e-mail: lixiaoping, wenbingtao@hust.edu.cn)



(a) Location Quality Distribution (b) Velocity Quality Distribution (c) Correlation

Fig. 2. Statistics of location and velocity quality distribution and their correlation on nuScenes *val* dataset. (a) and (b) reveal that the accuracy of all prediction locations and velocities are varying on an unfixed scale, showing irregularity. (c) shows a messy scatter plot, which reflects no relations between location and velocity results.

disturb the estimation performance. To further confirm this issue, we conduct an empirical analysis to study the predicted center location and velocity quality distribution as well as their correlations. Our study reveals two valuable points: (1) There exists a significant gap between the estimation error of 3D centers and that of velocities; (2) The predicted quality of location and velocity is extremely misaligned. The imbalanced tracking cues have little effect on the detection performance but play a dramatic role in MOT. The analysis cues motivate us to endow each predicted box with the self-diagnosis ability to track clues for realizing stable tracking association.

To this end, we propose to forecast the quality of tracking clues from the base 3D detector. Specifically, we introduce a Normalized Gaussian Quality (NGQ) metric with two dimensions to measure the quality of predicting center location and velocity. NGQ metric comprehensively considers the vector errors of the two predictions in a 2D vector space, which is a prerequisite for our tracking framework. Based on the quality estimation of NGQ, we design a robust association mechanism, i.e., the Quality-aware Object Association (QOA) strategy. It adopts the velocity quality to filter out low-quality motion candidates and leverages the location quality to further rule out the center positions of boxes with bad estimations. Therefore, QOA not only effectively deals with hard cases, but also avoids dangerous association. In a sense, our method is subordinate to the idea of "Put Quality Before Quantity" principle.

Through combining the proposed methods with the baseline 3D detector, we obtain a simple and robust 3D MOT framework, namely *quality-aware 3D tracker* (QTrack). We conduct extensive experiments on nuScenes dataset [8], showing significant improvements in the 3D MOT task. Comprehensively, the contributions of this work are summarized as follows:

- We conduct extensive empirical analysis to point out that the predicted quality of center location and velocity exists large distribution gap and misalignment relationship, making efficient CV tracking framework fall into sub-optimal performance.
- We first propose to predict the quality of velocity and location quality measured by the designed NGQ metric. Afterwards, we further introduce QOA to leverage the two qualities for ensuring safe association in 3D MOT task.
- QTrack achieves SOTA performance on nuScenes dataset which outperforms other camera-based methods by a large margin. Specially, QOA improves the baseline tracker by +2.2% AMOTA among several 3D detector settings, showing its effectiveness.

II. RELATED WORK

A. 3D Multi-object Tracking

Thanks to the development of 3D detection [9], [10], [11] and 2D MOT technologies [12], [4], [13], [14], recent 3D MOT methods [5], [7], [2], [3], [6] mainly follow tracking-by-detection paradigm. These trackers following this paradigm first utilize a 3D object detector to localize the targets in the 3D space (including location, rotation, and velocity) and then associate the detected objects with the trajectories according to various cues (location or appearance).

Traditional 3D MOT usually uses a motion model (Kalman filter) to predict the location of the tracklets and then associate the candidate detections using 3D (G)IoU [5], [6] or L2 distance [7]. Some works also utilize the advanced appearance model (ReID) [2], [15], [2] or temporal model (LSTM) [16], [3] to provide more reference cues for the association. Recently, Transformer [17] has been used in 3D detection [18] and MOT [19], [20] to learn 3D deep representations with 2D visual information and trajectory encoded. Although these methods achieved remarkable performance, when they are applied to complex scenarios (e.g., occlusion, motion blur, or light weakness), the tracking performance becomes unsatisfactory. In this work, we argue that a simple velocity clue with quality estimation can deal with the corner cases and achieve robust tracking performance. Our proposed QTrack focuses on how to assess the quality of the location and velocity prediction, and then make full use of these quality scores in the matching process.

B. Prediction Quality Estimation

Estimating the quality of the model's prediction is non-trivial, which can be applied to tackle prediction imbalance or decision-making. In the field of object detection, advanced works [21], [22], [23] introduce to predict a box's centerness or IoU for perceiving the quality of prediction (3D) boxes.

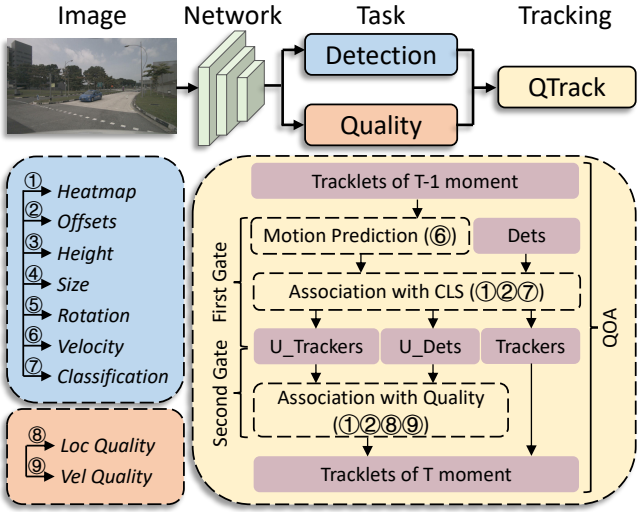


Fig. 3. Overview of base 3D detector and QTrack. The multi-view images are first fed into the detector, i.e., BEVDepth. Then we add two parallel branches for predicting location and velocity quality, respectively. For QTrack, it first employs velocity clue to conduct motion prediction, and then adopts heatmap, offsets, and classification score to carry out the association procedure in the first gate stage. Specially, location and velocity qualities are introduced to execute this work's key module QOA for unmatched trackers and detections in the second gate stage.

[24] employs the method to perceive the mask predicted quality. These methods can alleviate the imbalance between classification score and location accuracy. [25] introduces an uncertainty-based method to estimate the predicted quality of several depth factors, and then the quality is employed to make optimal decisions. In this paper, we introduce to predict the predicted quality of velocity and location. Afterwards, the predicted quality will be adopted to eliminate the non-robust association case of the tracking task. To our knowledge, our work is the first effort to perceive the quality velocity and location for the decision-making in 3D MOT task.

III. METHODOLOGY

A. Background of 3D Multiple Object Tracking (3D MOT)

3D MOT is a crucial task in computer vision, aiming to estimate the trajectories of multiple objects in 3D space over time. Formally, given current detection results $\{D_1, D_2, ..., D_n\}$, the goal of 3D MOT is to update history trajectories $\{T_1^{t-1}, T_2^{t-1}, ..., T_m^{t-1}\}$ by associating identity detection results. In addition, it needs to handle the disappearing objects for discarding the trajectories and the newly appearing objects for initializing the new trajectories.

B. Delve into the Quality Distribution

We aim to solve the task of 3D multi-object tracking (3D MOT), the goal of which is to locate the objects in the 3D space and then associate the detected targets with the same identity into the tracklets. The key challenge is how to associate the tracklets efficiently and correctly. In contrast to the motion-based and appearance-based association strategies, we argue that the simple velocity cue (CV method) is enough for the association, which is more lightweight and deployment-friendly. However, the performance of the existing CV tracking framework is not satisfactory. To analyze the reason for the limited performance of tracking with velocity, we count and visualize the distribution of the prediction error between location and velocity. As illustrated in Fig. 2 (a) and (b), we can observe that the distribution of the location and velocity quality (prediction error) is scattered, and a sizable number of low-quality boxes are included. Moreover, Fig. 2 (c) shows that the distribution correlation between the location and velocity error is nonlinear, which means the quality of the location and velocity is seriously misaligned.

Based on these observations, we conclude that the limited performance of tracking with velocity is due to the following reasons: (1) Low quality of the location or velocity. When one of the location and velocity predictions is not accurate enough, the tracker can not perform well even if the other prediction is reliable. (2) Misalignment between the quality of location and velocity. We should take both location and velocity quality into consideration. Driven by this analysis, we propose *Location and Velocity Quality Learning* to learn the quality uncertainty of the location, and velocity that can assist the tracker to select high-quality candidates for the association.

C. Base 3D Object Detector

Our method can be easily coupled with most existing 3D object detectors with end-to-end training. In this paper, we take BEVDepth [10] as an example. BEVDepth is a camera-based Bird's-Eye-View (BEV) 3D object detector that transfers the multi-view image features to the BEV feature through a depth estimation network and then localizes and classifies the objects in the BEV view. It consists of four kinds of modules: an image-view encoder, a view transformer with explicit depth supervision utilizing encoded intrinsic and extrinsic parameters, a BEV encoder and a task-specific head. The entire network is optimized with a multi-task loss:

$$L_{det} = L_{depth} + L_{cls} + L_{reg}, \tag{1}$$

where the depth loss L_{depth}, classification loss L_{cls} and regression loss L_{reg} remain the same setting as the original paper. As illustrated in Fig. 3, the task of the regression branch includes heatmap, offsets, height, size, rotation and velocity.

D. Location and Velocity Quality Learning

To effectively estimate the quality of location and velocity, it first needs to define the quality measurement metric. Technically, the box's center location is calculated by incorporating the predicted heatmap and corresponding offsets so that the location quality can be simplified to offset the predicted quality. Especially, the offsets and velocity are defined in a 2-dimensional vector space of Bird's Eye View (BEV). We introduce a Normalized Gaussian Quality (NGQ) metric to represent their quality. Given a predicted vector $\mathbf{P} \in \mathbb{R}^2$ and ground truth vector $\mathbf{G} \in \mathbb{R}^2$, we formulate NGQ metric as:

$$\mathrm{NGQ} = e^{-\frac{\sqrt{(\mathrm{P}x-\mathrm{G}x)^2+(\mathrm{P}y-\mathrm{G}y)^2}}{\gamma}}, \tag{2}$$

Algorithm 1: Pseudo-code of QOA.

Input: A video sequence V; object detector Det; detection score threshold τ; quality score threshold μ_v, μ_t
Output: Tracks T of the video

1 Initialization: $T \leftarrow \emptyset$
2 **for** *frame* f_k *in* V **do**
  /* boxes & scores */
3 $D_k \leftarrow \mathtt{Det}(f_k)$
4 $D_{high} \leftarrow \emptyset$
5 $D_{low} \leftarrow \emptyset$
  /* **first gate** */
6 **for** d *in* D_k **do**
7 **if** $d.score > \tau$ **then**
8 $D_{high} \leftarrow D_{high} \cup \{d\}$
9 **end**
10 **else**
11 $D_{low} \leftarrow D_{low} \cup \{d\}$
12 **end**
13 **end**
  /* predict location */
14 **for** t *in* T **do**
15 $t \leftarrow \mathtt{CV}(t)$
16 **end**
  /* association with high scores */
17 $W_{high} \leftarrow Cost(D^{(t)}_{high}, T^{(t-1)})$
18 **end**
19 with L2 distance
20 $D_{first}, T_{first} = Hungarian(W_{high})$
21 $D_{remain} \leftarrow D_{high} \setminus D_{first}$
22 $T_{remain} \leftarrow T \setminus T_{first}$
  /* association with low scores */
23 $W_{low} \leftarrow Cost(D^{(t)}_{low}, T_{remain})$
24 $D_{sec}, T_{sec} = Hungarian(W_{low})$
25 $T_{re-remain} \leftarrow T_{remain} \setminus T_{sec}$
  /* **second gate** */
26 **for** t, d *in* T_{sec}, D_{sec} **do**
27 **if** $t.v_{score} < \mu_v$ *or* $d.l_{score} < \mu_t$ **then**
28 $T_{re-remain} \leftarrow T_{re-remain} \cup \{t\}$
29 **end**
30 **end**
  /* update and initialize */
31 $T \leftarrow T \setminus T_{re-remain}$
32 **for** d *in* D_{remain} **do**
33 $T_{new} \leftarrow$ initialize track $\{d\}$
34 **end**
35
36 $T \leftarrow T \cup T_{new}$ Return: T

where the subscripts x and y indicate the value in the x and y directions while γ is a hyper-parameter to control the value distribution of NGQ. We set γ to 1.0 and 3.0 for location and velocity, respectively. **P** and **G** can be instantiated as predicting offset and velocity. When the prediction is equal to ground truth, NGQ = 1, while the predicted error is larger, NGQ is closer to 0.

After defining the quality, we elaborate on how to learn it. As shown in Fig. 3, we attach a 3×3 convolution layer for offset and velocity branch to predict location quality $\mathrm{NGQ}^{loc} \in \mathbb{R}^1$ and velocity quality $\mathrm{NGQ}^{vel} \in \mathbb{R}^1$, respectively. The quality supervision is conducted by binary cross entropy (BCE) loss:

$$L_{quality} = -\frac{1}{N}\sum_{i=1}^{N}[\hat{\mathrm{NGQ}}_\mathrm{i} \cdot \log \mathrm{NGQ}_\mathrm{i} + (1-\mathrm{NGQ}_\mathrm{i}) \cdot \log(1-\hat{\mathrm{NGQ}}_\mathrm{i})], \tag{3}$$

where $\hat{\mathrm{NGQ}}$ is the ground truth quality calculated by Eq. 2. This far, the total loss for our detector is formulated as:

$$L_{total} = L_{det} + L_{quality}. \tag{4}$$

The overall training procedure is in an end-to-end manner while the quality prediction task will not damage the performance of the base detector. Moreover, quality estimation is used in our proposed Quality-aware Object Association (QOA) module, which will be discussed next section.

E. Quality-aware Object Association

After obtaining the quality of the center location and velocity, we have more reference cues to achieve robust and accurate association. To this end, we propose a simple but effective quality-aware object association strategy (QOA). Specifically, QOA sets up two "gates". The first gate is the classification confidence score (cls score). We first separate the candidate detection boxes into high score ones and low score ones according to their cls scores. The high score candidates are first associated with the tracklets. Then the unmatched tracklets are associated with the low score candidates. These low score candidates are most caused by occlusion, motion blur, or light weakness, which are easily confused with the miscellaneous boxes. To deal with the issue, the second gate, quality uncertainty score, is introduced. After getting the second association results between the unmatched tracklets and the low score candidates, we then recheck the matched track-det pairs according to the location and velocity quality scores. Only high-quality matched track-det pairs can remain and low-quality pairs are regarded as the mismatch. The pseudo-code of QOA is shown in Algorithm 1.

Benefiting from the quality estimation, QOA does not need a complex motion or appearance model to provide association cues. A simple velocity prediction (CV) is enough (line #15). Hence, we use the velocity of the tracklet at frame $t-1$ to predict the center location at frame t and then compute the L2 distance between predictions and candidate detections (line #17 and line #20) as the similarity. At last, we apply the similarity with the Hungarian algorithm to get the association results. Mathematically,

$$\begin{aligned} c_t &= c_{t-1} + v_{t-1}\Delta t \\ cost &= L_2(c_t, d_t) \\ match &= Hungarian(cost), \end{aligned} \tag{5}$$

where c_{t-1}, v_{t-1} represents the center location and velocity of the tracklets at frame $t-1$. d_t is the candidate detection center location at frame t and Δt is the time lag.

IV. Experiments

A. Datasets and Metrics

Datasets. We mainly evaluate our QTrack on the 3D detection and tracking datasets of nuScenes. nuScenes dataset is a large-scale autonomous driving benchmark that consists of 1000 real-world sequences, 700 sequences for training, 150 for validation, and 150 for the test.

TABLE I

COMPARISON WITH STATE-OF-THE-ART METHODS ON NUSCENES VALIDATION AND TEST DATASET.

Split	Methods	Base Detector	Tracking Strategy	AMOTA↑	AMOTP↓	RECALL↑	MOTA↑	IDS↓	MOTP↓	FRAG↓	MT↑	ML↓
Val	DEFT [2]	-	Postprocessing	0.201	N/A	N/A	0.171	N/A	-	-	-	-
	QD3DT [3]	-	Postprocessing	0.242	1.518	39.9%	0.218	5646	-	-	-	-
	TripletTrack [16]	-	Postprocessing	0.285	1.485	N/A	N/A	N/A	-	-	-	-
	MUTR3D [20]	DETR3D	End-to-end	0.294	1.498	42.7%	0.267	3822	-	-	-	-
	PF-Track [26]	PETRv2	End-to-end	0.479	1.227	**59.0%**	0.435	**181**	-	-	-	-
	QTrack (Ours)	BEVDepth	Postprocessing	**0.511**	**1.090**	58.5%	**0.465**	1144	-	-	-	-
Test	CenterTrack	CenterNet	Postprocessing	0.046	1.543	23.3%	0.043	3807	0.753	2645	573	5235
	DEFT [2]	-	Postprocessing	0.177	1.564	33.8%	0.156	6901	0.770	3420	1951	3232
	Time3D [19]	-	End-to-end	0.210	1.360	N/A	0.173	N/A	N/A	N/A	N/A	N/A
	QD3DT [3]	Faster R-CNN	Postprocessing	0.217	1.550	37.5%	0.198	6856	0.773	3001	1893	2970
	TripletTrack [16]	-	Postprocessing	0.268	1.504	40.0%	0.245	1044	0.800	3978	2085	2922
	MUTR3D [20]	DETR3D	End-to-end	0.270	1.494	41.1%	0.245	6018	0.709	2749	2221	3133
	PolarDETR [27]	PolarDETR	End-to-end	0.273	1.185	40.4%	0.238	2170	0.719	1924	2266	3617
	SRCN3D [28]	SRCN3D	End-to-end	0.398	1.317	53.8%	0.359	4090	0.709	2769	2859	2278
	PF-Track [26]	PETRv2	End-to-end	0.434	1.252	53.8%	0.378	**249**	0.674	839	3548	2708
	Sparse4D [29]	Sparse4D	End-to-end	0.519	1.078	63.3%	0.459	1090	0.622	1525	4332	**2118**
	UVTR [30]	UVTR	Postprocessing	0.519	1.125	59.9%	0.447	2204	0.650	1949	3741	2236
	MV-ByteTrack	PETRv2	Postprocessing	0.564	1.005	63.5%	0.471	704	0.616	1016	4388	2278
	QTrack (Ours)	BEVDepth	Postprocessing	0.480	1.107	56.9%	0.431	1484	0.597	1356	3728	2540
	QTrack (Ours)	VideoBEV	Postprocessing	0.548	0.983	63.1%	**0.475**	1433	**0.554**	1169	4069	2337
	QTrack (Ours)	StreamPETR	Postprocessing	**0.566**	**0.975**	**65.0%**	0.460	784	0.576	**755**	**4533**	2218

Metrics. For 3D tracking task, we report average multi-object tracking accuracy (AMOTA) and average multi-object tracking precision (AMOTP). We also report metrics used in 2D tracking task from CLEAR [31], e.g., multi-object tracking accuracy (MOTA), multi-object tracking precision (MOTP), and number of identity switches (IDS).In addition, we report more metrics compared with other methods, e.g., the number of track fragmentations (Frag), the number of mostly tracked trajectories (MT), and the number of mostly lost trajectories (ML). To validate that the detection performance is not affected, we report nuScenes Detection Score (NDS), mean Average Prediction (mAP), as well as five True Positive (TP) metrics including mean Average Translation Error (mATE), mean Average Scale Error (mASE), mean Average Orientation Error (mAOE), mean Average Velocity Error (mAVE), mean Average Attribute Error (mAAE).

B. Implementation Details

Following BEVdepth, we adopt three types of backbone: ResNet-50 [32], ResNet-101, and VoVNet-99 (Initialized from DD3D [33]) as the image backbone. If not specified, the image size is processed to 256×704. The data augmentation includes random cropping, random scaling, random flipping, and random rotation. In addition, we also adopt BEV data augmentations including random scaling, random flipping, and random rotation. We use AdamW as optimizer with learning rate of 2×10^{-4} and batch size of 64. When compared with other methods, QTrack is trained for 24 epochs for ResNet and 20 epochs for VoVNet with CBGS [34]. As for VideoBEV and StreamPETR, we use the same settings from their submission version in the official test leaderboard. Both approaches utilize a Recurrent Neural Network (RNN) and adopt a stream query design, respectively, to process long sequences of images.

C. Comparision with Preceding SOTAs

Test and validation set. We compare the performance of QTrack with preceding SOTA methods on the nuScenes benchmark. The results are reported in Tab. 1. Our QTrack outperforms all current SOTA methods for the camera-based trackers by a large margin. For both validation and test sets, all reported metrics (e.g., AMOTA, AMOTP, RECALL, IDS, etc.) achieve best performance. Specially, AMOTA of QTrack first achieves 0.566, which significantly reduces the gap between pure camera and LiDAR-based trackers.

Compare with other post-processing trackers. Tab. II illustrates that QTrack outperforms the naive Kalman filter-based method and its advanced variant from SimpleTrack [6] by employing identical 3D detector and backbone settings. Tab. II also reports the latency triggered by 4 different post-processing tracking. It reflects that our method uses less latency or similar latency compared with SimpleTrack and KF models. Our method only needs simple operations (i.e., Matrix multiplication and addition) for tracking procedure, while Kalman filter-based ones need relatively complex operations like matrix inverse and the complex process for adjusting hyper-parameters. In addition, SimpleTrack needs to calculate complex GIoU for matching, which cost more inference time. The overall tracking framework is significantly efficient and will not trigger a serious latency, which is fatal in a real perception scenario [35], [36].

D. Ablation Study

In this subsection, we verify the effectiveness of the proposed strategies separately through ablation studies. All the experiments are conducted on the nuScenes *val* set.

Analysis of the components of QTrack. In this part, we verify the effectiveness of various components in QTrack through an ablation study. As shown in Tab. III, the first row of the table shows baseline performance for tracking when using BEVDepth detections followed by a simple velocity association step (CV method). We can observe that the two gates of QOA can both develop the tracking performance in the all settings (ResNet-50, ResNet-101 or VoVNet-99, single-frame or multi-frame), which means that the filter for the low-quality association results is necessary. Furthermore,

TABLE II

COMPARISON WITH DIFFERENT POST-PROCESSING TRACKERS ON NUSCENES VALIDATION DATASET. WE REPORT THE TRACKING RESULTS WITH TWO DIFFERENT BACKBONES, AND THE RESOLUTION OF THE INPUT IMAGE IS 256×704. THE LATENCY MEASURES THE PROCESSING COST OF THE TRACKING MODEL. THE LATENCY OF OUR METHOD FAIRLY INCLUDES THE EXTRA LIGHT-WEIGHT QUALITY PREDICTION HEAD.

Methods	Backbone	AMOTA ↑	AMOTP ↓	RECALL ↑	MOTA ↑	MOTP ↓	IDS ↓	FRAG ↓	MT ↑	ML ↓	Latency ↓
BEVDepth + KF	ResNet-50	0.303	1.337	39.7%	0.284	0.705	1290	780	1462	3344	3.14 ms
BEVDepth + CV	ResNet-50	0.325	1.276	42.8%	0.300	0.710	903	907	1843	3299	2.36 ms
BEVDepth + SimpleTrack	ResNet-50	0.338	1.294	43.9%	0.304	0.742	950	904	1798	3213	3.87 ms
BEVDepth + Ours	ResNet-50	0.347	1.347	42.6%	0.309	0.722	944	1106	1758	3137	3.24 ms
BEVDepth + KF	ResNet-101	0.301	1.345	40.2%	0.287	0.685	1444	841	1591	3156	3.14 ms
BEVDepth + CV	ResNet-101	0.323	1.282	42.1%	0.299	0.696	807	885	2359	3256	2.36 ms
BEVDepth + SimpleTrack	ResNet-101	0.333	1.302	42.4%	0.303	0.701	887	904	1835	3174	3.87 ms
BEVDepth + Ours	ResNet-101	0.339	1.349	42.8%	0.309	0.691	1100	1187	1956	2890	3.24 ms

TABLE III

ABLATION STUDY OF THE COMPONENTS IN QTRACK. CLS INDICATES THE FIRST GATE CLASSIFICATION SCORE WHILE Q. INDICATES THE SECOND GATE, I.E., QUALITY SCORE.

Backbone	MF	CLS	Q.	AMOTA↑	AMOTP↓	MOTA↑	IDS↓
ResNet-50	✗			29.1	1.314	26.7	1488
		✓		30.7	1.394	28.3	1748
		✓	✓	31.3	1.390	28.5	1559
	✓			32.5	1.276	30.0	903
		✓		34.1	1.348	30.5	1141
		✓	✓	34.7	1.347	30.9	944
ResNet-101	✗			29.1	1.314	26.7	1488
		✓		31.2	1.389	28.4	1622
		✓	✓	31.8	1.386	29.1	1638
	✓			32.3	1.282	30.9	1100
		✓		33.2	1.352	30.3	1053
		✓	✓	33.9	1.349	30.9	1100
VoVNet-99	✗			38.8	1.220	35.3	1670
		✓		40.4	1.266	36.9	1575
		✓	✓	40.8	1.259	37.0	1527
	✓			41.7	1.177	37.3	914
		✓		42.2	1.236	38.1	1125
		✓	✓	42.6	1.228	38.3	1076

TABLE IV

ABLATION STUDY OF HOW TO USE VELOCITY QUALITY (VQ) AND LOCATION QUALITY (LQ). THE VOVNET-99 BACKBONE IS USED.

MF	VQ	LQ	AMOTA↑	AMOTP↓	MOTA↑	IDS↓
✗			40.4	1.266	36.9	1575
	✓		40.1	1.269	36.6	1680
		✓	40.5	1.264	**37.3**	**1445**
	✓	✓	**40.8**	**1.259**	37.0	1527
✓			42.2	1.236	38.1	1125
	✓		41.9	1.239	38.0	**999**
		✓	42.2	1.235	38.2	1023
	✓	✓	**42.6**	**1.228**	**38.3**	1076

TABLE V

EFFECTIVENESS ON BOX ASSOCIATION METHOD.

Our method	AMOTA↑	AMOTP↓	MOTA↑	IDS↓
	30.3	1.337	0.284	1290
✓	31.4	1.282	0.297	1273

we can observe that the metric of IDS increases when applying the first gate by classification confidence score. This phenomenon shows that only considering confidence score inevitably introduces low-quality bounding boxes, which causes bad association cases. Therefore, the second gate, quality score, can provide a fine-grained reference to achieve a better association trade-off.

TABLE VI

INFLUENCE OF EXTRA BRANCH ON PERFORMANCE AND TRACKING DETECTION. FOR TRACKING PERFORMANCE OF CV AND SIMPLETRACK, WE REPORT THE AMOTA METRIC FOR COMPARISON.

Extra Branch	mAP↑	NDS↑	CV	SimpleTrack
None	0.3579	0.4826	0.326	0.337
Appearance	0.3522	0.4798	0.315	0.328
Relative Drop	-0.57%	-0.38%	-1.1%	-0.9%
Quality	0.3585	0.4831	0.325	0.338
Relative Drop	**+0.06%**	**+0.05%**	-0.1%	**+0.1%**

Analysis of the location and velocity quality for tracking. In this part, we conduct an in-depth analysis on the location and velocity quality score for the association process. As mentioned before, location and velocity quality scores are obtained by the quality branch. Then they are both regarded as the reference clues to filter the low classification confidence association results in QOA. We verify the performance of only using one of them as the second gate of QOA, and the results are reported in Tab. IV. As shown, only using one of the location and velocity quality scores does not contribute to the tracking performance, which confirms our analysis that the location and velocity quality is not aligned and we should take both of them into consideration.

Effectiveness on box association method. We have further expanded our approach to include the uncertainty of the 2D Bounding Box in Bird's Eye View (BEV) for box-association based tracking methods. Specifically, we consider the Intersection over Union (IoU) between predicted BEV 2D bounding boxes and Ground Truth (GT) ones as the uncertainty, denoted as U. We then apply the gating strategy outlined in Algorithm 1 for this box-association based tracking method. We modify line 25 of Algorithm 1 to "if $U < \mu_u$", which implies that the secondary association is not carried out if the box uncertainty is smaller than μ_u. Tab. V, which presents the results when our method is adapted to box-matching tracking methods.

Influence on base 3D detector. Tab. VI proves that adding quality prediction branch does not affect the performance of base 3D detector. This is an extremely important property since post-processing trackers normally rely on the super performance of detector. Going one step further, we report the tracking performance by employing existing CV and

SimpleTrack scheme. It reveals that tracking performance will not be affected by our quality branch, which agree with our designing purpose of Sec. 1. Then, we explore to append a appearance branch for extracting instance wised appearance embedding, which implement is the same as [37]. The results show that slight performance degradation (nearly 0.5%) is triggered on detection task, but it significantly damages the performance of tracking task by nearly 1.0%. It reflects that our method is more effective and efficient.

Qualitative results. We conducte a qualitative analysis comparing both CV method and our approach. This analysis specifically addresses a scenario where a car obscures a pedestrian, resulting in a lower classification score, and a person reflected in a mirror generates a false positive prediction, which has a higher classification score. Here are the responses from the three models:

1) **CV:** The true positive prediction is filtered out due to the low classification score caused by the occlusion, interrupting the trajectory.
2) **QTrack w/o quality:** Leveraging a two-gate strategy, the true positive prediction is given an opportunity to be associated again. However, the false positive prediction, which has a higher classification score, is mistakenly associated, resulting in a false trajectory.
3) **QTrack:** By predicting the quality based on velocity and location, the true positive prediction is assigned a higher quality score, while the false positive prediction receives a lower score. As a result, the false positive prediction is discarded.

Limitation. Albeit QTrack shows promising results in several key metrics, it still has some flaws: a) Our method requires the setting of an optimal factor, denoted as γ in Eq. 2, during the training process. The search for this optimal value may introduce additional time costs, which is a potential limitation. b) While QTrack excels in tracking the most crucial metric, AMOTA, it may not perform as well in other metrics. For instance, in terms of IDS for trajectory continuity, other trackers may show superior results.

E. Discussion and Future Work

Inspired by [23], [38], [39], we explore incorporating velocity quality V with classification score C as M, which is adopted to act as a threshold metric in NMS procedure. We use the new metric M to act as the judgment standard to remove the weight of the overlapping box. Technically, we formulate M in Eq. 6, in which α is a hyper-parameter to control the contribution of V.

$$M = V^{(1-\alpha)} \cdot C^{\alpha}, \tag{6}$$

As shown in Fig. 4, we plot the four performance metrics of the detection task by controlling α. It reflects that as the contribution of V becomes bigger (i.e., enlarge $1-\alpha$), mAVE drops dramatically, which means the velocity prediction accuracy becomes better. However, it also brings about inevitable performance degradation for mAP and mATE metrics. NDS, as a comprehensive metric, becomes better

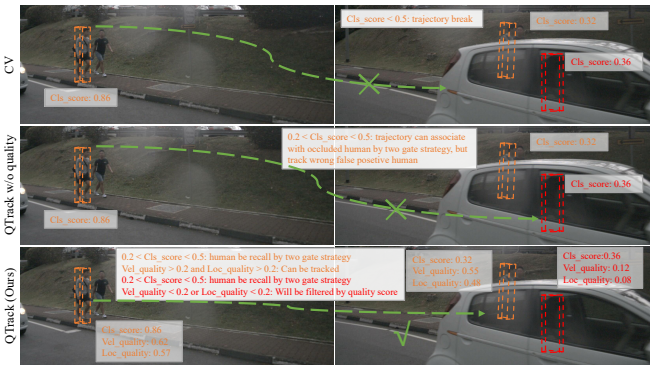


Fig. 4. Qualitative results for 3D MOT.

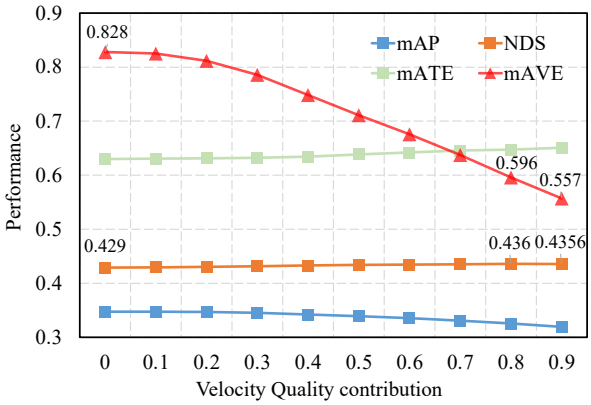


Fig. 5. Influence of velocity quality on detection performance in NMS procedure. The abscissa indicates α in Eq. 6.

and then gets worse as α changes larger, which is actually a trade-off between location error and velocity error. This phenomenon agrees with our viewpoint in Sec. 1, i.e., the quality of these two prediction tasks is not aligned. Combining the performance of detection and tracking tasks with respect to above imbalance issue, it exposes a challenge: *how to design a method to simultaneously predict location (or 3D box) and velocity well?* This challenge can help further boost the performance of 3D detection task or other downstream tasks like 3D MOT.

V. CONCLUSIONS

In this paper, we analyze the imbalanced prediction quality distribution of location and velocity. It motivates us to propose a Quality-aware Object Association (QOA) method to alleviate the imbalance issue for 3D multi-object tracking (3D MOT). To this end, we introduce Normalized Gaussian Quality (NGQ) metric to measure the predicted quality of location and velocity, and structure an effective module for quality learning. Afterwards, we further present QTrack, an "tracking by detection" framework for 3D MOT in multi-view camera scene, which incorporats with QOA to perform tracking procedure. The extensive experiments demonstrate the efficacy and robustness of our method. Finally, we release a challenge to inspire more research to focus on the imbalance between localization and velocity qualities for both 3D detection and tracking tasks.

REFERENCES

[1] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering 82(Series D)*, pages 35–45, 1960.

[2] Mohamed Chaabane, Peter Zhang, J Ross Beveridge, and Stephen O'Hara. Deft: Detection embeddings for tracking. *arXiv preprint arXiv:2102.02267*, 2021.

[3] Hou-Ning Hu, Yung-Hsu Yang, Tobias Fischer, Trevor Darrell, Fisher Yu, and Min Sun. Monocular quasi-dense 3d object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[4] En Yu, Zhuoling Li, Shoudong Han, and Hongwei Wang. Relationtrack: Relation-aware multiple object tracking with decoupled representation. *IEEE Transactions on Multimedia*, 2022.

[5] Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. 3d multi-object tracking: A baseline and new evaluation metrics. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10359–10366. IEEE, 2020.

[6] Ziqi Pang, Zhichao Li, and Naiyan Wang. Simpletrack: Understanding and rethinking 3d multi-object tracking. *arXiv preprint arXiv:2111.09621*, 2021.

[7] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11784–11793, 2021.

[8] 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.

[9] Junjie Huang, Guan Huang, Zheng Zhu, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021.

[10] Yinhao Li, Zheng Ge, Guanyi Yu, Jinrong Yang, Zengran Wang, Yukang Shi, Jianjian Sun, and Zeming Li. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. *arXiv preprint arXiv:2206.10092*, 2022.

[11] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. *arXiv preprint arXiv:2203.05625*, 2022.

[12] Shoudong Han, Piao Huang, Hongwei Wang, En Yu, Donghaisheng Liu, and Xiaofeng Pan. Mat: Motion-aware multi-object tracking. *Neurocomputing*, 476:75–86, 2022.

[13] En Yu, Zhuoling Li, and Shoudong Han. Towards discriminative representation: Multi-view trajectory contrastive learning for online multi-object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8834–8843, 2022.

[14] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[15] Xinshuo Weng, Yongxin Wang, Yunze Man, and Kris M Kitani. Gnn3dmot: Graph neural network for 3d multi-object tracking with 2d-3d multi-feature learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6499–6508, 2020.

[16] Nicola Marinello, Marc Proesmans, and Luc Van Gool. Triplettrack: 3d object tracking using triplet embeddings and lstm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4500–4510, 2022.

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[18] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on Robot Learning*, pages 180–191. PMLR, 2022.

[19] Peixuan Li and Jieyu Jin. Time3d: End-to-end joint monocular 3d object detection and tracking for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3885–3894, 2022.

[20] Tianyuan Zhang, Xuanyao Chen, Yue Wang, Yilun Wang, and Hang Zhao. Mutr3d: A multi-camera tracking framework via 3d-to-2d queries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4537–4546, 2022.

[21] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 913–922, 2021.

[22] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019.

[23] Borui Jiang, Ruixuan Luo, Jiayuan Mao, Tete Xiao, and Yuning Jiang. Acquisition of localization confidence for accurate object detection. In *Proceedings of the European conference on computer vision (ECCV)*, pages 784–799, 2018.

[24] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring r-cnn. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6409–6418, 2019.

[25] Zhuoling Li, Zhan Qu, Yang Zhou, Jianzhuang Liu, Haoqian Wang, and Lihui Jiang. Diversity matters: Fully exploiting depth clues for reliable monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2791–2800, 2022.

[26] Ziqi Pang, Jie Li, Pavel Tokmakov, Dian Chen, Sergey Zagoruyko, and Yu-Xiong Wang. Standing between past and future: Spatio-temporal modeling for multi-camera 3d multi-object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17928–17938, 2023.

[27] Shaoyu Chen, Xinggang Wang, Tianheng Cheng, Qian Zhang, Chang Huang, and Wenyu Liu. Polar parametrization for vision-based surround-view 3d detection. *arXiv preprint arXiv:2206.10965*, 2022.

[28] Yining Shi, Jingyan Shen, Yifan Sun, Yunlong Wang, Jiaxin Li, Shiqi Sun, Kun Jiang, and Diange Yang. Srcn3d: Sparse r-cnn 3d surround-view camera object detection and tracking for autonomous driving. *arXiv preprint arXiv:2206.14451*, 2022.

[29] Xuewu Lin, Tianwei Lin, Zixiang Pei, Lichao Huang, and Zhizhong Su. Sparse4d: Multi-view 3d object detection with sparse spatial-temporal fusion. *arXiv preprint arXiv:2211.10581*, 2022.

[30] Yanwei Li, Yilun Chen, Xiaojuan Qi, Zeming Li, Jian Sun, and Jiaya Jia. Unifying voxel-based representation with transformer for 3d object detection. *Advances in Neural Information Processing Systems*, 35:18442–18455, 2022.

[31] Keni Bernardin, Alexander Elbs, and Rainer Stiefelhagen. Multiple object tracking performance metrics and evaluation in a smart room environment. In *Sixth IEEE International Workshop on Visual Surveillance, in conjunction with ECCV*, volume 90. Citeseer, 2006.

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[33] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is pseudo-lidar needed for monocular 3d object detection? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3142–3152, 2021.

[34] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced grouping and sampling for point cloud 3d object detection. *arXiv preprint arXiv:1908.09492*, 2019.

[35] Jinrong Yang, Songtao Liu, Zeming Li, Xiaoping Li, and Jian Sun. Real-time object detection for streaming perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5385–5395, 2022.

[36] Jinrong Yang, Songtao Liu, Zeming Li, Xiaoping Li, and Jian Sun. Streamyolo: Real-time object detection for streaming perception. *arXiv preprint arXiv:2207.10433*, 2022.

[37] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, 129(11):3069–3087, 2021.

[38] Shengkai Wu, Xiaoping Li, and Xinggang Wang. Iou-aware single-stage object detector for accurate localization. *Image and Vision Computing*, 97:103911, 2020.

[39] Jinrong Yang, Shengkai Wu, Lijun Gou, Hangcheng Yu, Chenxi Lin, Jiazhuo Wang, Pan Wang, Minxuan Li, and Xiaoping Li. Scd: A stacked carton dataset for detection and segmentation. *Sensors*, 22(10):3617, 2022.