

# Time3D: End-to-End Joint Monocular 3D Object Detection and Tracking for Autonomous Driving

Peixuan Li  
SAIC PP-CEM

lpeixuan@saicmotor.com

Jieyu Jin  
SAIC PP-CEM

jinjieyu@saicmotor.com

## Abstract

While separately leveraging monocular 3D object detection and 2D multi-object tracking can be straightforwardly applied to sequence images in a frame-by-frame fashion, stand-alone tracker cuts off the transmission of the uncertainty from the 3D detector to tracking while cannot pass tracking error differentials back to the 3D detector. In this work, we propose jointly training 3D detection and 3D tracking from only monocular videos in an end-to-end manner. The key component is a novel spatial-temporal information flow module that aggregates geometric and appearance features to predict robust similarity scores across all objects in current and past frames. Specifically, we leverage the attention mechanism of the transformer, in which self-attention aggregates the spatial information in a specific frame, and cross-attention exploits relation and affinities of all objects in the temporal domain of sequence frames. The affinities are then supervised to estimate the trajectory and guide the flow of information between corresponding 3D objects. In addition, we propose a temporal-consistency loss that explicitly involves 3D target motion modeling into the learning, making the 3D trajectory smooth in the world coordinate system. Time3D achieves 21.4% AMOTA, 13.6% AMOTP on the nuScenes 3D tracking benchmark, surpassing all published competitors, and running at 38 FPS, while Time3D achieves 31.2% mAP, 39.4% NDS on the nuScenes 3D detection benchmark.

## 1. Introduction

3D object detection is an essential task for Autonomous Driving. Compared with LiDAR system, monocular cameras are cheap, stable, and flexible, favored by mass-produced cars [4, 9, 18, 19]. However, monocular 3D object detection is a natural ill-posed problem for the lack of depth information, making it difficult to estimate an accurate and stable state of the 3D target [19, 22]. A typical solution is to smooth the previous and current state through 2D multiple

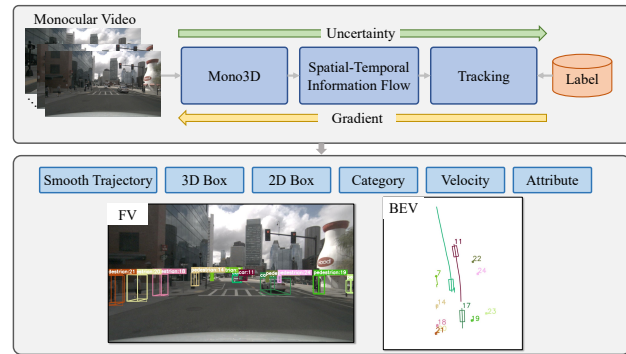


Figure 1. **Illustration of the proposed Time3D.** Given a monocular video sequence, Time3D joint learn monocular 3D object detection and 3D tracking, and output smooth trajectories, 2D boxes, 3D boxes, category, velocity, and motion attribute. Time3D is trained end-to-end, so it can feed forward the uncertainty and backward the error gradient.

object trackers(MOT) [5, 20].

Following the “tracking-by-detection” paradigm [2, 30, 37], a widely used contemporary strategy first computes past trajectory representations and detected objects. Later, that association module is used to compute the similarity between the current objects across the past frames to estimate their tracks. Most of the existing methods following this pipeline try to build several different cues, including re-identification(Re-ID) models [25, 40, 41], motion modeling [12, 15], and hybrid models [1]. Currently, most of these models are still hand-crafted so that the corresponding tracker can only follow the detector independently. The recent studies of 2D MOT try to establish a deep learning association in tracking [3, 29, 43]. However, these methods still encounter three shortcomings in autonomous driving scenarios: **1)** They treat detection and association separately, in which stand-alone tracking module cuts off the transmission of the uncertainty from the 3D detector to tracking while cannot differential pass error back to the 3D detector. **2)** The objects from the same category often have similar appearance information and often undergo frequent

occlusions and different speed variations in the autonomous driving scenario. They failed to integrate these heterogeneous cues in a unified network. **3)** They estimate the trajectory without directly constraining the flow of appearance and geometric information in the network, which is crucial to the trajectory smoothness, velocity estimation, and motion attribute (e.g., parking, moving, or stopped for cars).

We propose to combine 3D monocular object detection and 3D MOT into a unified architecture with end-to-end training manner, which can: **(1)** predict 2D box, 3D box, Re-ID feature from only monocular images without any extra synthetic data, CAD model, instance mask, or depth map. **(2)** encode the compatible feature representation for these cues. **(3)** learn a differential association to generate trajectory by simultaneously combining heterogeneous cues across time. **(4)** guide the information flowing across all objects to generate a target state with temporal consistency. To do so, we first modify the anchor-free monocular 3D detector KM3D [18] to learn the 3D detector and Re-ID embedding jointly, following the "joint detection and tracking" paradigm [34, 42], so that it can generate 2D box, 3D box, object category, and Re-ID features, simultaneously. To design the compatible feature representation for different cues, we propose to transform the parameters of the different magnitude of the 2D box and 3D box to the unified representation, 2D corners, and 3D corners, in which the geometric information can be extracted as a high-dimensional feature from corner raw coordinates by the widely used PointNet [24] structure. Fig. 1 illustrate the pipeline of Time3D.

Reviewing MOT, we found that data association is very similar to the Query-Key mechanism, where one object is a query, and another object in a different frame is the key. Its feature in different frames is highly similar for the same object, enabling the query-key mechanism to output a high response. We, therefore, propose a Transformer architecture, a widely-used entity of query-key mechanism. Inspired by RelationNet [14], the self-attention aggregates features from all elements in a frame to exploit spatial topology, which is automatically learned without any explicit supervision. The cross-attention computes the target affinities across different frames, and its query-key weights are supervised to learn tracks via a unimodal loss function. The finally temporal-spatial features output the velocity, attribute, and 3D box smoothness refinement. In addition, we propose temporal-consistency loss that constrains the temporal topology of objects in the 3D world coordinate system to make the trajectory smoother.

To summarize, the main contributions in this work are the following: **(1)**. We propose a unified framework to jointly learn 3D object detection and 3D multi-object tracking by combining heterogeneous cues in an end-to-end manner. **(2)**. We propose an embedding extractor to make

geometric and appearance information compatible by transforming the 2D and 3D boxes to unified representations. **(3)**. We propose a temporal-consistency loss to make the trajectory smoother by constraining the temporal topology. **(4)**. Experiments in the nuScenes 3D tracking benchmark demonstrate that the proposed method achieves the best tracking accuracy comparing other competitors by large margins while running in real-time (26FPS).

## 2. Related Work

**Monocular 3D Object Detection** *Monocular 3D object detection* is a naturally ill-posed problem that is more complicated than 2D detection. The central problem is the lack of depth from only the perspective images. To address this challenge, the variants of Pseudo-LiDAR [33] first estimate a depth map and then transform the depth map into point clouds following a LiDAR-based methods [38, 39, 45] to detect 3D objects. Another method [21] remove the transforming step and directly use depth to guide estimation in CNN. Inspired by the 2D anchor-based detection [4, 5], put the 3D anchors in 3D space, and filter anchors with the 2D box helping. In order to avoid direct estimation of the depth value, the variants of RTM3D [18, 19] tried to combine the CNN and geometry projection to infer depth or position. Although these methods perform well on a single image, autonomous driving scenarios are often sequential recognition tasks, and processing objects independently may cause sub-optimal results. For this reason, [5, 13] introduce MOT module to monocular 3D object detection to uses temporal information to predict stable 3D information. [5] takes kinematic motion into the Kalman Filter as the main tracking module, which independently runs the detection and tracking model, and only the hand-crafted geometric features are used for data association. [13] include more geometric information and appearance for data association, however, it leverages a hand-crafted data association strategy and not contain a spatial-temporal information flow for box consistency constrain, which is more important to downstream tasks of autonomous driving.

**Multi-Object Tracking** *Multi-object tracking* has been explored extensively in the 2D scope, in which most trackers follow the tracking-by-detection paradigm [11, 17, 20]. These models first predict all objects by a powerful 2D detector [26, 27, 44] in each frame, and then link them up by data association strategy. [20] associate each box through the Kalman filter with the distance metric of box IoU. [36] add the appearance information from a deep network to [20] for a more robust association, especially for occlusion objects. More recent approaches [28, 30, 37] follow these two methods and focus on to increase the robustness of data association. These approaches separate tracking into detection and data association stages, preventing the uncertainty flow and end-to-end training. A recent trend in MOT is to

reformulate existing trackers into the combination of both tasks in the same framework. Sun et al. [29] use a siamese network with the paired frame as input and predict the similarity score of detections. Guillem [3] proposes a graph partitioning method to treat the association problem as an edge classification problem. The disadvantages of these models are that they treat detection and association separately, learn limited cues, have a complicated structure, and are not practical in autonomous driving scenes. We jointly learn 3D detection and 3D tracking by exploiting heterogeneous cues while running in real-time.

### 3. Proposed Approach

Given a monocular image  $I_t$  at time  $t$  an autonomous vehicle needs to perceive the location  $P_t = \{P_t^i \in \mathbb{R}^3 | i = 1, \dots, n_t\}$ , dimension  $D_t = \{D_t^i \in \mathbb{R}^3 | i = 1, \dots, n_t\}$ , orientation  $R_t = \{R_t^i \in \mathbb{R}^3 | i = 1, \dots, n_t\}$ , velocity  $V_t = \{V_t^i \in \mathbb{R}^3 | i = 1, \dots, n_t\}$ , attribute  $A_t = \{A_t^i \in \mathbb{R}^3 | i = 1, \dots, n_t\}$ , and smooth trajectory  $\mathcal{T}_t = \{\mathcal{T}_t^i | i = 1, \dots, n_t\}$  of  $n_t$  objects around the scene to make motion planning and control.

Fig. 2 illustrate the whole architecture details of Time3D. Tim3D only takes monocular video images as input and consists of the following steps: **1).** An fast and accuracy **Monocular 3D Object Detector** in JDE mode [34] is designed to obtain 2D boxes, 3D boxes, category, and Re-ID embedding for every frame. **2).** An **Heterogeneous Cues Embedding** module that encodes appearance and geometric features to compatible feature representation. **3).** A **Spatial-Temporal Information Flow** module that propagates the information of all objects across frames to each other estimates the similarity to generate 3D trajectory and aggregates the geometry relative relationship in the world coordinate system to estimate velocity, attribute, and box smoothness refinement.

#### 3.1. Monocular 3D Object Detection

Our monocular 3D object detector takes an image  $I_t$  at certain time  $t$  as input, and outputs objects 2D box  $b_t^i = \{x_t^i, y_t^i, w_t^i, h_t^i\}, i = 1, \dots, n_t$ , 3D box  $B_t^i = \{X_t^i, Y_t^i, Z_t^i, W_t^i, H_t^i, L_t^i\}, i = 1, \dots, n_t$ , category  $Cls_{i=Car, Pedestrian, \dots}$ , and ReID embedding  $ReID_t^i$ . Specifically, we employ KM3D [18] as our monocular 3D detector, which predicts dimension, orientation, and nine perspective corners following a differentiable geometric reasoning module (GRM) for position estimation. KM3D is a camera-independent method that can handle a variety of cameras with different intrinsic and viewing characteristics, making it more practical. Following the FairMOT [42], we add a Re-ID head paralleled to other detection heads, which focuses on generating distinguish features of different objects. We implement the same convolution layer with other

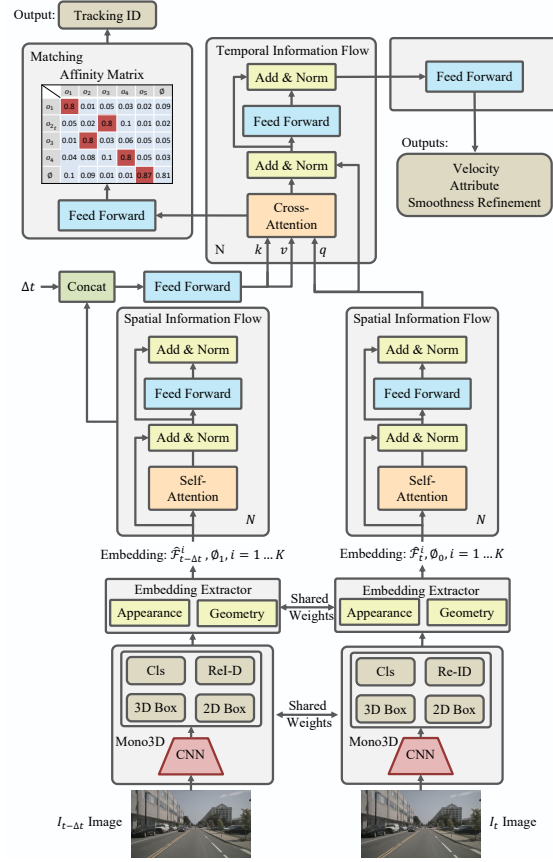


Figure 2. **The architecture details of Time3D.** First, the current and previous frame images are input to Mono3D to estimate Top  $K$  objects with category, 2D Box, 3D Box, and Re-ID feature. Then, the current and previous cues are fed into the extractor of heterogeneous cues embedding to generate appearance and geometric embedding. Next, learning objects embedding are propagated to each other in a spatial domain through spatial information flow. Finally, the temporal information flow matches the same object across frames to compute affinity matrix for estimation of tracks, while output velocity, motion attribute, and box smoothness refinement.

heads but output a 256 dimension vector to extract Re-ID features in each object’s 2D center.

#### 3.2. Heterogeneous Cues Embedding

An ideally data association is expected to extract embedding of multiple cues (e.g., appearance and geometric) over a long period. However, the appearance features (e.g. Re-ID feature) are in vector space, the geometric features (e.g. location, dimension, and orientation) are in Euclidean space, making them difficult to be combined in a unified network. As a result, only one of them was used in previous approaches [12, 15, 25, 40, 41], leading to sub-optimal results. In this paper, we elegantly encode the compatible representations of appearance, geometric, and motion in-

formation.

Specifically, for every 2D box  $b_t^i$  and 3D box  $B_t^i$  in image  $I_t$ , we first transform their parameters to the 2D corners  $C_2(b_t^i) \in \mathbb{R}^{4 \times 2}$  and 3D corners  $C_3(B_t^i) \in \mathbb{R}^{8 \times 3}$ . These corners are flattened and then fed into light-weight PointNet [24] structure, that compose of only 3 layers MLP and *MaxPooling* to generate geometric  $G_t^i \in \mathbb{R}^d$  features with  $d$  feature dimension.

In addition to Re-ID in the appearance features, we also add category cues, which can be further used to constrain the similarity of the same object between different frames. We present the category information from an MLP layer applied on the one-hot feature layer of the cls head in our monocular 3D detector. Then we simple add the category feature and Re-ID  $ReID_t^i$  to generate appearance feature  $A_t^i$ .

Motion modeling is time-aware processing that depends on the relative time between frames. We put motion modeling behind the spatial information flow in order to reduce calculation. We detail it in Sec. 3.3 and Sec. 3.5.

### 3.3. Spatial-Temporal Information Flow

This section reviews the transformer structure first and then introduces how to apply it to spatial-temporal information flow.

Transformer [31] was first introduced as a new network based on attention mechanisms for machine translation and then was formulated to process computer vision tasks by ViT [10] and DETR [7]. For a query  $\mathcal{Q}$ , key  $\mathcal{K}$ , and value  $\mathcal{V}$ , we simple present the multi-head attention as:

$$Y = MultiHeadAttn(\mathcal{Q}, \mathcal{K}, \mathcal{V}, PE) \quad (1)$$

where  $PE$  is the positional encoding function to eliminate the permutation-invariant. This attention is called self-attention if the input query and key are the same, otherwise cross-attention. The transformer stacks self-attention, normalization, and feed-forward layers in the encoder and decoder, and the cross-attention focuses on the interaction between them. We refer the reader to the literature [31] for more detailed descriptions.

The transformer architecture can be naturally extended to the spatial-temporal information, in which the self-attention propagate the objects' information at a certain time, and cross-attention aggregate objects information across times. The structure of the spatial information flow is shown at the bottom of Fig. 2. We first extract top- $K$  center points in image  $I_t$  from main center head of our 3D detector, and index its corresponding appearance features  $A_t^i$ , and geometric  $G_t^i$  (details in Sec. 3.2) following the concatenation with a MLP layer to generate input embedding  $\mathcal{F}_t^i$ .  $K$  is set to be larger than the typical number  $n_t$  of objects in an image( e.g. 128 in nuScenes dataset). So that

the spatial information flow can be summarized as:

$$\hat{\mathcal{F}}_t^i = MultiHeadAttn(\mathcal{F}_t^i, \mathcal{F}_t, \mathcal{F}_t, \mathbf{0}) \quad (2)$$

We set the positional encoding as  $\mathbf{0}$  for the reason that the geometric feature already contains the position information. The exact structure with different weights can be applied in another frame  $I_{t-\Delta t}$  to generate its propagation feature  $\hat{\mathcal{F}}_{t-\Delta t}$ .

The spatial information flow module process strictly follows self-attention to propagation information and encodes the spatial topology, which has been proved to be adequate to improve object detection by RelationNet [14].

The structure of the temporal flow module is shown at the top of Fig. 2. The temporal flow module aggregates the information from paired frame  $I_{t-\Delta t}$  and  $I_t$  by using multi-head cross-attention in the form of residual. In cross-attention, the dot-product weight explores the relationship of paired detection objects in different frames. It represents the consistency in probabilistic from 0-1 normalized by *softmax*, where 0 is defined as different objects, and 1 is the same object. This probabilistic naturally contains matching information and can be used as a similarity score for tracking directly. In order to prevent the ID switch, we design Time3D as a semi-global association, so we also need to capture time information. We simple *concat* the relative time  $\Delta t$  and  $\hat{\mathcal{F}}_{t-\Delta t}$  following a FFN layer to generate time-aware propagation feature  $\tilde{\mathcal{F}}_{t-\Delta t}$ , that also called motion modeling. We formulate the generation of affinity matrix as:

$$\Gamma(i, j)_t^{t-\Delta t} = FFN \left( W_q^i \hat{\mathcal{F}}_t^i * W_k^j \tilde{\mathcal{F}}_{t-\Delta t}^j / \sqrt{C} + \mathbf{0} \right) \quad (3)$$

where  $W_q$ , and  $W_k$  are of learnable weights.  $\Gamma(i, j)_t^{t-\Delta t}$  is affinity score between  $i^{th}$  object in frame  $I_t$  and  $j^{th}$  object in frame  $I_{t-\Delta t}$ . Similar to spatial information flow module, temporal positional encoding is set as  $\mathbf{0}$ . Considering that the objects in the image pair may not have a corresponding relationship, we learn a extra row and column for un-identified targets, following the DAN [29]. We simple add a FFN to weights of cross-attention and estimate affinity matrix  $\Gamma_t^{t-\Delta t} \in \mathbb{R}^{K+1 \times K+1}$ , which will be trained as a unimodal matrix for one-to-one matching.

The temporal information flow branch can generate tracking information and guide the temporal aggregation of appearance and geometric information. The temporal aggregation module then models the temporal transition of targets to predict box smoothness refinement and the time-related variable(e.g., *velocity*, *motion attribute*). Thus, the mechanism of temporal information flow can be summarized as:

$$\tilde{\mathcal{F}}_t^i = MultiHeadAttn(\hat{\mathcal{F}}_t^i, \tilde{\mathcal{F}}_{t-\Delta t}, \tilde{\mathcal{F}}_{t-\Delta t}, \mathbf{0}) \quad (4)$$



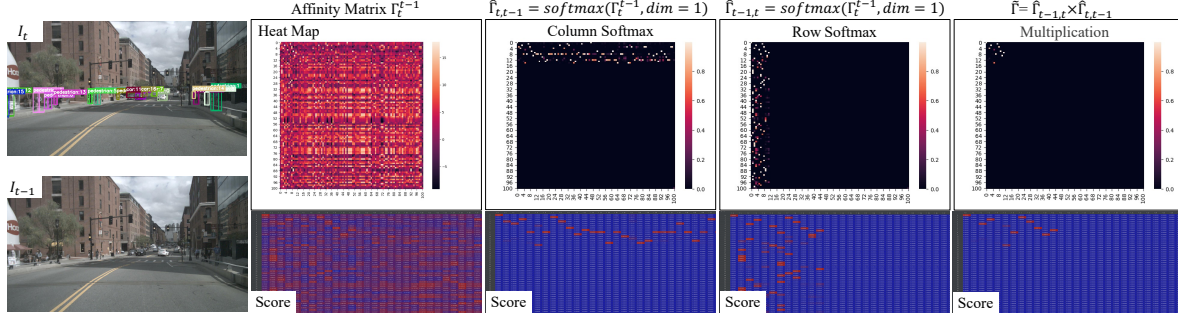


Figure 3. **Learnt unimodal affinity matrix.** Our simple tracking loss can learn a unimodal affinity matrix, which can perform a one-to-one matching.

where  $\tilde{\mathcal{F}}_t^i \in \mathbb{R}^{K \times d}$  is the aggregation features of  $i^{\text{th}}$  target at time  $t$ . The final prediction is computed by a FFN which consist of 3-layer MLP. The FFN predicts the velocity  $\mathbf{V}_t = \{\mathbf{V}_t^i \in \mathbb{R}^3 | i = 1, \dots, K\}$ , motion attribute  $\mathbf{M}_t = \{\mathbf{M}_t^i \in \mathbb{R}^3 | i = 1, \dots, K\}$ , and box smoothness refinement value  $\Delta \mathcal{B} = \{\Delta \mathcal{B}_t^i = \Delta X_t^i, \Delta Y_t^i, \Delta Z_t^i, \Delta W_t^i, \Delta H_t^i, \Delta L_t^i | i = 1, \dots, K\}$ .

### 3.4. Training Loss

We divide the multi-task loss into three parts: monocular object 3D detection loss  $\mathcal{L}_{Mono3D}$ , the tracking loss  $\mathcal{L}_{\mathcal{T}}$ , and temporal-consistency loss  $\mathcal{L}_{Cons}$ .

1). **Monocular object 3D detection loss.** We adopt the same loss as KM3D [18]:

$$\mathcal{L}_{Mono3D} = \mathcal{L}_m + \mathcal{L}_{kc} + \mathcal{L}_D + \mathcal{L}_O + \mathcal{L}_T \quad (5)$$

where the  $\mathcal{L}_m, \mathcal{L}_{kc}, \mathcal{L}_D, \mathcal{L}_O, \mathcal{L}_T$  are main center loss, key-point loss, dimension loss, orientation loss and position loss respectively.

2). **Tracking loss.** Benefit to our explicit modeling of the target appearance features and geometric features in spatial-temporal space, we can design a simpler and more effective loss function than DAN [29] to constrain the affinity matrix to generate a unimodal response. Specifically, we first perform row- and column-wise *softmax* to affinity matrix  $\Gamma_t^{t-\Delta t} \in \mathbb{R}^{K+1 \times K+1}$ , and results association matrix  $\hat{\Gamma}_{t,t-\Delta t}$  and  $\hat{\Gamma}_{t-\Delta t,t}$ . The  $i^{\text{th}}$  row of the matrix  $\hat{\Gamma}_{t,t-\Delta t}$  associates the  $i^{\text{th}}$  objects in frame  $I_t$  to  $K+1$  identities in frame  $I_{t-\Delta t}$ , where  $+1$  presents the un-identified targets in frame  $I_{t-\Delta t}$ . The matrix  $\hat{\Gamma}_{t-\Delta t,t}$  signify similar associations from  $I_{t-\Delta t}$  to  $I_t$ . We then multiply these two matrices  $\hat{\Gamma}_t^{t-\Delta t} = \Gamma_{t,t-\Delta t} \times \hat{\Gamma}_{t-\Delta t,t}$  and use cross-entropy losses for training:

$$\mathcal{L}_{tracking} = - \sum_{j=1}^K \sum_{i=1}^{K+1} * \Gamma_t^{t-\Delta t}(i, j) \log(\tilde{\Gamma}_t^{t-\Delta t}(i, j)) \quad (6)$$

where  $* \Gamma_t^{t-\Delta t}$  presents the ground truth association matrix, and  $* \Gamma_t^{t-\Delta t}(i, j) = 1$  indicates that the  $i^{\text{th}}$  identities in the

frame  $I_{t-\Delta t}$  and the  $j^{\text{th}}$  object in the frame  $I_t$  are the same object. Fig. 3 show a sample frames from nuScenes video to illustrate the processing of tracking loss.

3). **Temporal-consistency loss.** The design of most functions hopes to check the stability in temporal to ensure a smooth transition. However, previous approaches calculated detection results independently for each frame and failed to constrain a trajectory’s inner consistency in the network. We propose an auxiliary loss to clarify temporal topology between frames during the training phase to ensure trajectory smoothness. Specifically, we first add the smoothness refinement box to the outputs of our monocular 3D detector to generate final box parameters and then calculate the relative 3D corner distance of the same object in a different frame. The ground truth then supervises these corner distances:

$$\mathcal{L}_{Cons} = \sum_{\zeta=1}^{\tau} \sum_{i=0}^{n_t} \sum_{j=0}^{n_{t-\zeta}} \left| \left| \mathcal{C}_3(\mathbf{B}_t^i + \Delta \mathbf{B}_t^i) - \mathcal{C}_3(\mathbf{B}_{t-\zeta}^j + \Delta \mathbf{B}_{t-\zeta}^j) \right|_2 - \left| \mathcal{C}_3(\mathbf{B}_t^*) - \mathcal{C}_3(\mathbf{B}_{t-\zeta}^*) \right|_2 \right|_2 \quad (7)$$

where  $\mathcal{C}_3$  map the parameters of the 3D box to 8 corners.  $\mathbf{B}^*$  presents ground truth. Interestingly, although every potential target is supervised by corresponding ground truth, the temporal-consistency loss can still minimize the relative location. Fig. 6 shows a sample to explain that.

Finally, the Time3D is supervised in an end-to-end fashion with the multi-task loss combination:

$$\mathcal{L} = \mathcal{L}_{Mono3D} + \mathcal{L}_{Tracking} + \mathcal{L}_{Cons} \quad (8)$$

### 3.5. Inference of Tracking

The inference of tracking is shown in Fig. 4. We first perform the 3D object detection, heterogeneous cues extractor, and spatial information flow sequentially at each timestamp. Next, the spatial features are stored with their time stamps. Given the current frame image  $I_t$  with its spatial features, the affinity matrix is computed by forward passing

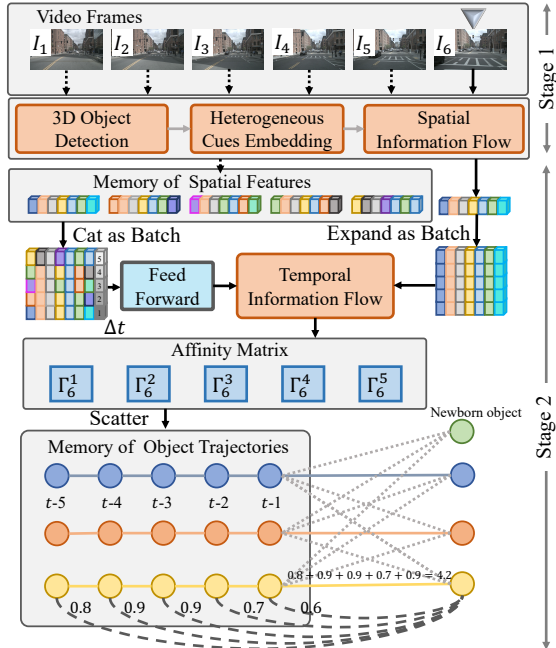


Figure 4. **Inference of tracking.** Each frame of images is first fed into 3D object detection, embedding extractor, and spatial information flow module to generate spatial features. We then equip an explicit memory to store the past trajectories with corresponding spatial features. Given the current spatial features, only the fast and straightforward temporal information flow is used to compute the similarity score.

through temporal information flow. In order to reduce the ID switch, we follow the DAN [29] to calculate the affinity matrix of the objects in the current frame and the spatial features stored in all the trajectories and sum them as the similarity score between the objects and the trajectories. Finally, the Hungarian algorithm [23] is adopted to get the optimal assignments. When running the assignment process frame-by-frame, object trajectories are yielded. Thus each frame image is passed through a high-weight network of 3D detection, embedding extractor, and spatial information only once, but the stored spatial features are used multiple times through light-weight temporal information flow to compute similarity score. Time3D, therefore, can run in real-time.

## 4. Experiments

### 4.1. Dataset

We evaluate Time3D on a large-scale popular nuScenes [32] dataset for autonomous driving. nuScenes is composed of 1000 scenes video, with 700/150/150 scenes for *train*, *validation*, and *testing* set, respectively. nuScenes collected data from Boston and Singapore, in both day and night, and different weather conditions. Each video contains images from six cameras forming an entire 360° field of view in a 2HZ keyframe sample. Finally, 1.4M 3D

bounding box annotations from 10 categories are contained. Considering that nuScenes includes sequence monocular images, 3D annotations, tracking ID, and the pose of each frame simultaneously, we use it as the benchmark to validate the performance of our method.

### 4.2. Implementation Details

In mono3D, we follow the KM3D [18] to report the results with the DLA-34 backbone. We stack three self-attention layers in spatial information flow, and stack four cross-attention layers in temporal information flow, in which we compute affinity matrix in  $2^{th}$  cross-attention without *softmax*. Time3D is trained without the KM3D pre-trained model but with ImageNet pre-trained model as initialization. We use the augmentation of shifting and scaling. We resize the image from  $900 \times 1600$  to  $448 \times 800$  for fast training. The batch size is 80 in 8 2080Ti GPUs, with ten images per GPU. We train Time3D for 200 epochs with a starting learning rate of  $1.25e4$  and reduce it by  $\times 10$  at epoch 90 and 120. We only associate the past five frames to the current frame for the fast running speed.

### 4.3. Comparison with state-of-the-art

We report four official evaluation metrics [15] for tracking quantitative analyses: Average Multi-Object Tracking Accuracy (AMOTA), Average Multi-Object Tracking Precision (AMOTP), Multi-Object Tracking Accuracy (MOTA), and Multi-Object Tracking Precision (MOTP), while reporting the seven official evaluation metrics for 3D detection: Average Precision (AP), Average Translation Error (ATE), Average Scale Error (ASE), Average Orientation Error (AOE), Average Velocity Error (AVE), Average Attribute Error (AAE), and nuScenes detection score (NDS). We only compare the online (no peeking into the future) tracking methods from published approaches for a fair comparison. As shown in Tab. 1, Time3D outperforms all the other online trackers with different metrics while maintaining outstanding running speed. As we all know, the MOT performance is highly relay on the accuracy of detection. We therefore list the results of two popular LiDAR-based 3D detector [16, 46] with baseline tracker AB3DMOT [35]. Megvii achieves high detection accuracy by employing LiDAR point cloud, but Time3D outperforms the Megvii by a large margin for 3D tracking, in which AMOTA improves 41.7%, AMOTP of 9%, MOTA 11%. We also perform a Time3D with no end-to-end training, in which we separately train KM3D, Re-ID extractor, and spatial-temporal module, detailed in supplementary. End-to-end Time3D improves 37% for AMOTA and 22% for MOTA. All these shows that the power of Time3D, which jointly learns detection and tracking in an end-to-end manner, can outperform those methods that use detection and tracking as two different steps, even if these methods have more powerful de-

Table 1. **3D Tracking and 3D Object Detection Performance nuScenes test set.** † represent it use the AB3DMOT [35] as the tracker. Time3D‡ trained in no end-to-end manner.

Method	Modality	Time	3D Multi-Object Tracking				3D Object Detection						
			AMOTA (%) ↑	AMOTP (m) ↓	MOTA (%) ↑	MOTP (m) ↓	mAP (%) ↑	mATE (m) ↓	mASE (1-iou) ↓	mAOE (rad) ↓	mAVE (m/s) ↓	mAAE (1-acc) ↓	NDS (%) ↑
Megvii† [6]	LiDAR	-	15.1	1.50	15.4	<b>0.40</b>	<b>52.8</b>	<b>0.30</b>	<b>0.25</b>	<b>0.38</b>	<b>0.25</b>	14.0	<b>63.3</b>
PointPillar† [6]	LiDAR	-	2.9	1.70	4.5	0.82	30.5	0.52	0.29	0.50	0.32	37.0	45.3
CenterNet [44]	Camera	-	-	-	-	-	33.8	0.66	0.26	0.63	1.63	14.2	40
FCOS3D [32]	Camera	-	-	-	-	-	35.8	0.69	0.25	0.45	1.43	12.4	42.8
CenterTrack [43]	Camera	45ms	4.6	1.54	4.3	0.75	-	-	-	-	-	-	-
DEFT [8]	Camera	-	17.7	1.56	15.6	0.77	-	-	-	-	-	-	-
MonoDIS† [6]	Camera	-	1.8	1.79	2.0	0.90	30.4	0.738	0.263	0.546	1.553	13.4	38.4
Time3D‡	Camera	50ms	15.6	1.49	14.1	0.78	32.9	0.716	0.250	0.511	1.647	14.8	39.9
Time3D	Camera	<b>38ms</b>	<b>21.4</b>	<b>1.36</b>	<b>17.3</b>	0.75	31.2	0.732	0.254	0.504	1.523	<b>12.1</b>	39.4

tection accuracy. Tab. 1 show Megvii has a smaller MOTP to Time3D, which is because LiDAR-based Megvii has a higher accuracy to a translation error.

Time3D is proposed to combine 3D detection and 3D tracking in a unified framework and make autonomous driving more practical. We, therefore, employ a lightweight monocular 3D detection method, KM3D, for the trade-off between speed and accuracy. The 3D detection accuracy of Time3D still has a gap to LiDAR-based methods Megvii and has the competitiveness to single task methods CenterNet [44] and FCOS3D [32]. The use of a more powerful 3D detector can improve the performance of 3D detection, and we will take it as future work. Note that Time3D‡ achieves higher accuracy in the 3D detection benchmark, and we will analyze this phenomenon in the ablation study.

#### 4.4. Qualitative Results

Fig. 5 shows some qualitative results of Time3D. Time3D can predict exciting tracking effects and output smooth trajectories, even for occluded or high-speed moving targets. It is worth noting that Time3D only uses monocular images as input and outputs 3D trajectories. Small targets such as pedestrians have relatively rough trajectories related to the 3D detector. It can be replaced with a more robust 3D detector to generate a more accurate 3D position. More qualitative results can be seen in the supplementary.

#### 4.5. Ablation Study

For a fair comparison, all experiments are trained in *training* set and tested in *val* set following the nuScenes official setting.

Table 2. **Ablation for different cues** Performance on nucenes *val*. \* represent directly encoding to 2D or 3D box.

Cls	2D	3D	Re-ID	AMOTA	AMOTP	MOTA	MOTP
✓				2.4	1.79	2.7	0.74
	✓			8.9	1.68	10.1	0.76
		✓		11.2	1.61	12.8	0.76
			✓	16.5	1.54	15.3	0.82
			✓	24.7	1.41	19.6	0.82
	✓	✓	✓	25.8	1.39	20.4	0.83
✓	✓	✓	✓	26.0	1.38	20.7	0.82
✓	✓*	✓*	✓	19.0	1.48	17.1	0.82

**Heterogeneous Cues Embedding.** We first examine the effects of different cues, such as Re-ID features, the embedding of 2D box, 3D box, and category. As shown in Tab 2, for the single cue testing, using only Re-ID features achieves the highest 16.5 AMOTA. Adding the other cues can further boost the performance. The best performance is achieved by combining all heterogeneous cues. We observe that the biggest improvement occurs in adding the 3D Box cue, which demonstrates that the 3D location is important to 3D object tracking. We expect to inform future research to focus on 3D information to improve the tracking performance further.

The geometric embedding extractor learns a compatible feature representation for 2D boxes and 3D boxes. We investigate its effect by comparing the solutions that directly encode 4 Dof and 7 Dof parameters of the 2D Box and 3D Box with the same MLP layer as the proposed geometric embedding extractor. As shown in the last row of Tab 2, direct encoding has dropped performance significantly, which demonstrates that an explicit geometry encoding method is more conducive to the learning of the network.

Table 3. **Ablation for with or without Re-ID feature.** Performance on nucenes *val*.

Setting	mAP	mATE	mASE	mAOE	NDS
w/	29.1	0.79	0.24	0.48	39.0
w/o	32.2	0.75	0.26	0.49	39.5

In addition, we observe that MOTP has a little worse when introducing the Re-ID feature. We, therefore, performed the detection experiment and found that introducing the Re-ID harms the detection performance. The results are shown in Tab. 3. This also explains that the detection accuracy of Time3D‡ is better than Time3D. The degeneration may be caused by the contradiction between the invariance of Re-ID "identity" and the variance of detection.

Table 4. **Ablation for Spatial-Temporal Information Flow**

Setting	AMOTA	AMOTP	MOTA	MOTP
w/o Spatial-Temporal	17.9	1.43	19.0	0.81
w/ Spatial	23.9	1.27	14.7	0.82
w/ Saptial-Temporal	26.0	1.38	20.7	0.82

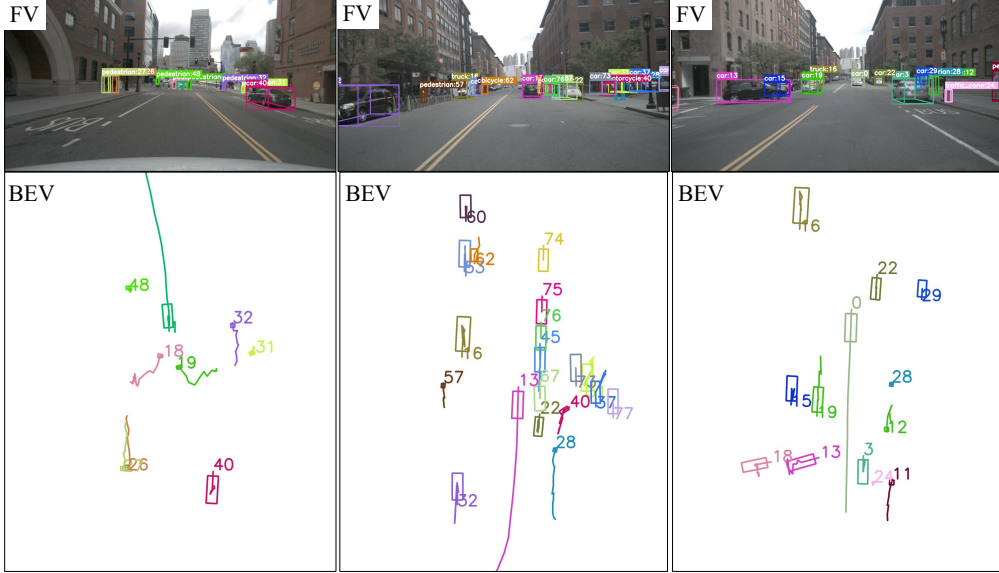


Figure 5. **Qualitative Results.** We visualize the tracking ID, 2D box, 3D box, category in front view in  $1^{th}$  row. The  $2^{th}$  row of Bird’s eye view (BEV) show the 3D box with the center point trajectory of the past 15 frames. Different colors represent objects with different tracking IDs.

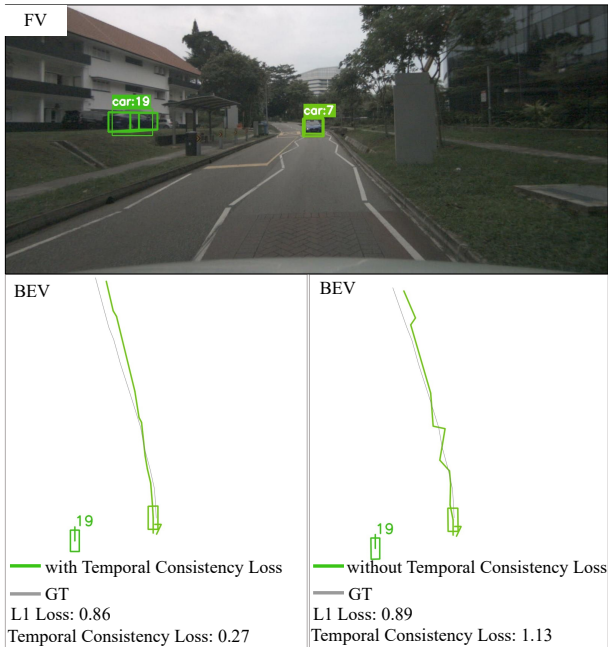


Figure 6. **A sample to illustrate the temporal-consistency loss.** We show two results predicted by Time3D with and without temporal-consistency loss. They have the same corner  $L_1$  loss but have different temporal-consistency loss. Time3D with temporal-consistency loss generates a smoother trajectory.

### Spatial-Temporal Information Flow.

Tab. 4 examines the effects of the spatial-temporal information flow module in improving the tracking accuracy. To avoid the influence caused by additional network parameters, we replace the spatial and temporal information mod-

ule with 6 MLP layers and cosine similarity. The proposed spatial-temporal information flow module improves by 45% in terms of AMOTA. At the same time, it can also predict velocity, motion attribute, and box smoothness refinement, indicating the *spatial-temporal information flow* can more explicitly detect the spatial-temporal topological relationship between objects.

**Temporal-Consistency Loss.** We show a sample in Fig. 6 to illustrate the effect of temporal-consistency loss. The trajectory with temporal consistency-loss is smoother than the trajectory without temporal-consistency loss. They have a similar  $L_1$  mean distance loss to the ground truth but have different temporal-consistency loss. We propose the temporal-consistency matrices to evaluate the smoothness of trajectory. The details can be seen in supplementary.

## 5. Conclusion

This work proposes a novel framework to jointly learn 3D object detection and 3D multi-object tracking from only monocular video while running in real-time. Our framework encodes the heterogeneous cues, including category, 2D Box, 3D Box, and Re-ID feature, to compatible embedding. The transformer-based architecture performs spatial-temporal information flow to estimation trajectory, optimized by temporal-consistency loss to smoother. On the nuScenes dataset, the proposed Time3D achieves state-of-the-art tracking performance while running in real-time. The Time3D may inspire future researchers to combine 3D tracking and 3D detection in a unified framework, and to encode more 3D information to make vision-based autonomous driving more practical.



## References

- [1] Seung Hwan Bae and Kuk-Jin Yoon. Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(3):595–610, 2018. [1](#)
- [2] Alex Bewley, ZongYuan Ge, Lionel Ott, Fabio Tozeto Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing, ICIP 2016, Phoenix, AZ, USA, September 25-28, 2016*, pages 3464–3468. IEEE, 2016. [1](#)
- [3] Guillem Brasó and Laura Leal-Taixé. Learning a neural solver for multiple object tracking. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 6246–6256. Computer Vision Foundation / IEEE, 2020. [1](#), [3](#)
- [4] Garrick Brazil and Xiaoming Liu. M3d-rpn: Monocular 3d region proposal network for object detection. In *Proceedings of the IEEE International Conference on Computer Vision, Seoul, South Korea, 2019*. [1](#), [2](#)
- [5] Garrick Brazil, Gerard Pons-Moll, Xiaoming Liu, and Bernt Schiele. Kinematic 3d object detection in monocular video. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXIII*, volume 12368 of *Lecture Notes in Computer Science*, pages 135–152. Springer, 2020. [1](#), [2](#)
- [6] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenet: A multi-modal dataset for autonomous driving. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 11618–11628. Computer Vision Foundation / IEEE, 2020. [7](#)
- [7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *ECCV (1)*, volume 12346 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2020. [4](#)
- [8] Mohamed Chaabane, Peter Zhang, J. Ross Beveridge, and Stephen O’Hara. DEFT: detection embeddings for tracking. *CoRR*, abs/2102.02267, 2021. [7](#)
- [9] Mingyu Ding, Yuqi Huo, Hongwei Yi, Zhe Wang, Jianping Shi, Zhiwu Lu, and Ping Luo. Learning depth-guided convolutions for monocular 3d object detection. In *CVPR Workshops*, pages 4306–4315. IEEE, 2020. [1](#)
- [10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. [4](#)
- [11] Kuan Fang, Yu Xiang, Xiaocheng Li, and Silvio Savarese. Recurrent autoregressive networks for online multi-object tracking. In *2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018, Lake Tahoe, NV, USA, March 12-15, 2018*, pages 466–475. IEEE Computer Society, 2018. [2](#)
- [12] Zeyu Fu, Pengming Feng, Syed Mohsen Naqvi, and Jonathon A. Chambers. Particle PHD filter based multi-target tracking using discriminative group-structured dictionary learning. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5-9, 2017*, pages 4376–4380. IEEE, 2017. [1](#), [3](#)
- [13] Hou-Ning Hu, Qi-Zhi Cai, Dequan Wang, Ji Lin, Min Sun, Philipp Krähenbühl, Trevor Darrell, and Fisher Yu. Joint monocular 3d vehicle detection and tracking. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 5389–5398. IEEE, 2019. [2](#)
- [14] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. *CoRR*, abs/1711.11575, 2017. [2](#), [4](#)
- [15] Tino Kutschbach, Erik Bochinski, Volker Eiselein, and Thomas Sikora. Sequential sensor fusion combining probability hypothesis density and kernelized correlation filters for multi-object tracking in video data. In *14th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2017, Lecce, Italy, August 29 - September 1, 2017*, pages 1–5. IEEE Computer Society, 2017. [1](#), [3](#), [6](#)
- [16] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 12697–12705. Computer Vision Foundation / IEEE, 2019. [6](#)
- [17] Laura Leal-Taixé, Cristian Canton-Ferrer, and Konrad Schindler. Learning by tracking: Siamese CNN for robust target association. In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2016, Las Vegas, NV, USA, June 26 - July 1, 2016*, pages 418–425. IEEE Computer Society, 2016. [2](#)
- [18] Peixuan Li. Monocular 3d detection with geometric constraints embedding and semi-supervised training. 2020. [1](#), [2](#), [3](#), [5](#), [6](#)
- [19] Peixuan Li, Huaici Zhao, Pengfei Liu, and Feidao Cao. Rtm3d: Real-time monocular 3d detection from object keypoints for autonomous driving, 2020. [1](#), [2](#)
- [20] Keng-Chi Liu, Yi-Ting Shen, and Liang-Gee Chen. Simple online and realtime tracking with spherical panoramic camera. In *IEEE International Conference on Consumer Electronics, ICCE 2018, Las Vegas, NV, USA, January 12-14, 2018*, pages 1–6. IEEE, 2018. [1](#), [2](#)
- [21] Xinzhu Ma, Shinan Liu, Zhiyi Xia, Hongwen Zhang, Xingyu Zeng, and Wanli Ouyang. Rethinking pseudo-lidar representation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *ECCV (13)*, volume

- 12358 of *Lecture Notes in Computer Science*, pages 311–327. Springer, 2020. [2](#)
- [22] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3d bounding box estimation using deep learning and geometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7074–7082, 2017. [1](#)
- [23] J. Munkres. Algorithms for the assignment and transportation problems. *SIAM J.*, 10, 1962. [6](#)
- [24] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation, 2016. cite arxiv:1612.00593. [2](#), [4](#)
- [25] Hossein Rahmani, Ajmal S. Mian, and Mubarak Shah. Learning a deep model for human action recognition from novel viewpoints. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(3):667–681, 2018. [1](#), [3](#)
- [26] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. [2](#)
- [27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. [2](#)
- [28] Sarthak Sharma, Junaid Ahmed Ansari, J. Krishna Murthy, and K. Madhava Krishna. Beyond pixels: Leveraging geometry and shape cues for online multi-object tracking. In *2018 IEEE International Conference on Robotics and Automation, ICRA 2018, Brisbane, Australia, May 21-25, 2018*, pages 3508–3515. IEEE, 2018. [2](#)
- [29] ShiJie Sun, Naveed Akhtar, HuanSheng Song, Ajmal Mian, and Mubarak Shah. Deep affinity network for multiple object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(1):104–119, 2021. [1](#), [3](#), [4](#), [5](#), [6](#)
- [30] Siyu Tang, Mykhaylo Andriluka, Bjoern Andres, and Bernt Schiele. Multiple people tracking by lifted multicut and person re-identification. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 3701–3710. IEEE Computer Society, 2017. [1](#), [2](#)
- [31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv*, 2017. [4](#)
- [32] 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 (ICCV) Workshops*, pages 913–922, October 2021. [6](#), [7](#)
- [33] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8445–8453, 2019. [2](#)
- [34] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards real-time multi-object tracking. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XI*, volume 12356 of *Lecture Notes in Computer Science*, pages 107–122. Springer, 2020. [2](#), [3](#)
- [35] Xinshuo Weng and Kris Kitani. A baseline for 3d multi-object tracking. *CoRR*, abs/1907.03961, 2019. [6](#), [7](#)
- [36] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing, ICIP 2017, Beijing, China, September 17-20, 2017*, pages 3645–3649. IEEE, 2017. [2](#)
- [37] Jiarui Xu, Yue Cao, Zheng Zhang, and Han Hu. Spatial-temporal relation networks for multi-object tracking. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 3987–3997. IEEE, 2019. [1](#), [2](#)
- [38] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10), 2018. [2](#)
- [39] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018. [2](#)
- [40] Shengping Zhang, Xiangyuan Lan, Hongxun Yao, Huiyu Zhou, Dacheng Tao, and Xuelong Li. A biologically inspired appearance model for robust visual tracking. *IEEE Trans. Neural Networks Learn. Syst.*, 28(10):2357–2370, 2017. [1](#), [3](#)
- [41] Xingming Zhang, Xuehan Zhang, Xuedan Du, Xiangming Zhou, and Jun Yin. Learning multi-domain convolutional network for RGB-T visual tracking. In Wei Li, Qingli Li, and Lipo Wang, editors, *11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics, CISP-BMEI 2018, Beijing, China, October 13-15, 2018*, pages 1–6. IEEE, 2018. [1](#), [3](#)
- [42] Yifu Zhang, Chunyu Wang, Xinggong Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *Int. J. Comput. Vis.*, 129(11):3069–3087, 2021. [2](#), [3](#)
- [43] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part IV*, volume 12349 of *Lecture Notes in Computer Science*, pages 474–490. Springer, 2020. [1](#), [7](#)
- [44] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. In *arXiv preprint arXiv:1904.07850*, 2019. [2](#), [7](#)
- [45] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. 2017. [2](#)
- [46] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced grouping and sampling for point cloud 3d object detection. *CoRR*, abs/1908.09492, 2019. [6](#)