Track to Detect and Segment: An Online Multi-Object Tracker

Jialian Wu[1], Jiale Cao[2], Liangchen Song[1], Yu Wang[3], Ming Yang[3], Junsong Yuan[1]

[1]SUNY Buffalo [2]TJU [3]Horizon Robotics

Abstract

Most online multi-object trackers perform object detection stand-alone in a neural net without any input from tracking. In this paper, we present a new online joint detection and tracking model, TraDeS (TRAck to DEtect and Segment), exploiting tracking clues to assist detection end-to-end. TraDeS infers object tracking offset by a cost volume, which is used to propagate previous object features for improving current object detection and segmentation. Effectiveness and superiority of TraDeS are shown on 4 datasets, including MOT (2D tracking), nuScenes (3D tracking), MOTS and Youtube-VIS (instance segmentation tracking). Project page: https://jialianwu.com/projects/TraDeS.html.

1. Introduction

Advanced online multi-object tracking methods follow two major paradigms: tracking-by-detection [5, 38, 27, 52, 30, 49] and joint detection and tracking [26, 63, 1, 29, 45, 25, 43, 44]. The tracking-by-detection (*TBD*) paradigm treats detection and tracking as two independent tasks (Fig. 1 (a)). It usually applies an off-the-shelf object detector to produce detections and employs another separate network for data association. The *TBD* system is inefficient and not optimized end-to-end due to the two-stage processing. To address this problem, recent solutions favor a joint detection and tracking (*JDT*) paradigm that simultaneously performs detection and tracking in a single forward-pass (Fig. 1 (b)).

The *JDT* methods, however, are confronted with two issues: **(i)** Although in most *JDT* works [29, 45, 25, 50] the backbone network is shared, detection is usually performed standalone without exploring tracking cues. We argue that detection is the cornerstone for a stable and consistent tracklet, and in turn tracking cues shall help detection, especially in tough scenarios like partial occlusion and motion blur. **(ii)** As studied by [9] and our experiment (Tab. 1b), common re-ID tracking loss [45, 25, 32, 51] is not that compatible with detection loss in jointly training a single backbone network, which could even hurt detection performance to some extent.

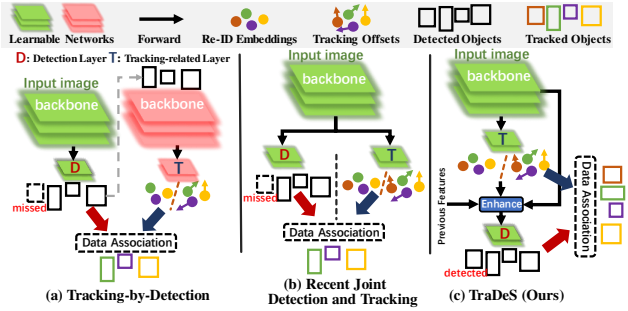

Figure 1. **Comparison of different online MOT pipelines.** Our method follows the joint detection and tracking (*JDT*) paradigm. Different from most *JDT* methods, the proposed TraDeS tracker deeply couples tracking and detection within an end-to-end and unified framework, where the motion clue from tracking is exploited to enhance detection or segmentation (omitted in the figure).

The reason is that re-ID focuses on intra-class variance, but detection aims to enlarge inter-class difference and minimize intra-class variance.

In this paper, we propose a new online joint detection and tracking model, coined as TraDeS (TRAck to DEtect and Segment). In TraDeS, each point on the feature map represents either an object center or a background region, similar as in CenterNet [64]. TraDeS addresses the above two issues by tightly incorporating tracking into detection as well as a dedicatedly designed re-ID learning scheme. Specifically, we propose a cost volume based association (CVA) module and a motion-guided feature warper (MFW) module, respectively. The CVA extracts point-wise re-ID embedding features by the backbone to construct a cost volume that stores matching similarities between the embedding pairs in two frames. Then, we infer the tracking offsets from the cost volume, which are the spatio-temporal displacements of all the points, *i.e.,* potential object centers, in two frames. The tracking offsets together with the embeddings are utilized to conduct a simple two-round long-term data association. Afterwards, the MFW takes the tracking offsets as motion cues to propagate object features from the previous frames to the current one. Finally, the propagated feature and the current feature are aggregated to derive detection and segmentation.

In the CVA module, the cost volume is employed to su-

pervise the re-ID embedding, where different object classes and background regions are implicitly taken into account. This is being said, our re-ID objective involves the inter-class variance. This way not only learns an effective embedding as common re-ID loss [45, 25, 32, 51], but also is well compatible with the detection loss and does not hurt detection performance as shown in Tab. 1b. Moreover, because the tracking offset is predicted based on appearance embedding similarities, it can match an object with very large motion or in low frame rate as shown in Fig. 3, or even accurately track objects in different datasets with unseen large motion as shown in Fig. 4. Thus, the predicted tracking offset of an object can serve as a robust motion clue to guide our feature propagation in the MFW module. The occluded and blurred objects in the current frame may be legible in early frames, so the propagated features from previous frames may support the current feature to recover potentially missed objects by our MFW module.

In summary, we propose a novel online multi-object tracker, TraDeS, that deeply integrates tracking cues to assist detection in an end-to-end framework and in return benefits tracking as shown in Fig. 1 (c). TraDeS is a general tracker, which is readily extended to instance segmentation tracking by adding a simple instance segmentation branch. Extensive experiments are conducted on 4 datasets, *i.e.*, MOT, nuScenes, MOTS, and Youtube-VIS datasets, across 3 tasks including 2D object tracking, 3D object tracking, and instance segmentation tracking. TraDeS achieves state-of-the-art performance with an efficient inference time as shown in § 5.3. Additionally, thorough ablation studies are performed to demonstrate the effectiveness of our approach as shown in § 5.2.

2. Related Work

Tracking-by-Detection. MOT was dominated by the tracking-by-detection (*TBD*) paradigm over the past years [58, 6, 66, 52, 33, 5, 38, 48, 54]. Within this framework, an off-the-shelf object detector [31, 16] is first applied to generate detection boxes for each frame. Then, a separate re-ID model [1, 49] is used to extract appearance features for those detected boxes. To build tracklets, one simple solution is to directly compute appearance and motion affinities with a motion model, *e.g.,* Kalman filter, and then solve data association by a matching algorithm. Some other efforts [6, 46, 19] formulate data association as a graph optimization problem by treating each detection as a graph node. However, *TBD* methods conduct detection and tracking separately, hence are usually computationally expensive. Instead, our approach integrates tracking cues into detection and efficiently performs detection and tracking in an end-to-end fashion.

Joint Detection and Tracking. Recently joint detection and tracking (*JDT*) paradigm has raised increasing attention due to its efficient and unified framework. One common way [63, 45, 25, 1, 62, 61] is to build a tracking-related branch upon an object detector to predict either object tracking offsets or re-ID embeddings for data association. Alternatively, transformer is exploited to match tracklets [36, 26]. CTracker [29] constructs tracklets by chaining paired boxes in every two frames. TubeTK [28] directly predicts a box tube as a tracklet in an offline manner. Most *JDT* methods, however, are confronted with two issues: First, detection is still separately predicted without the help from tracking. Second, the re-ID loss has a different objective from that of detection loss in joint training. In contrast, our TraDeS tracker addresses these two problems by tightly incorporating tracking cues into detection and designing a novel re-ID embedding learning scheme.

Tracking-guided Video Object Detection. In video object detection, a few attempts [15, 62] exploit tracking results to reweight the detection scores generated by an initial detector. Although these works strive to help detection by tracking, they have two drawbacks: First, tracking is leveraged to help detection only at the post-processing stage. Detections are still predicted by a standalone object detector, so detection and tracking are separately optimized. Thus, the final detection scores may heavily rely on the tracking quality. Second, a hand-crafted reweighting scheme requires manual tune-up for a specific detector and tracker. Our approach differs from these post-processing methods because our detection is learned conditioned on tracking results, without a complex reweighting scheme. Therefore, detection tends to be robust *w.r.t.* tracking quality.

Cost Volume. The cost volume technique has been successfully applied in depth estimation [11, 55, 18] and optical flow estimation [35, 10, 53] for associating pixels between two frames. This motivates us to extend cost volume to a multi-object tracker, which will be demonstrated to be effective in learning re-ID embeddings and inferring tracking offsets in this paper. Our approach may inspire future works using cost volume in tracking or re-identification.

3. Preliminaries

The proposed TraDeS is built upon the point-based object detector CenterNet [64]. CenterNet takes an image $\boldsymbol{I} \in \mathbb{R}^{H\times W\times 3}$ as input and produces the base feature $\boldsymbol{f} = \phi(\boldsymbol{I})$ via the backbone network $\phi(\cdot)$, where $\boldsymbol{f} \in \mathbb{R}^{H_F\times W_F\times 64}$, $H_F = \frac{H}{4}$, and $W_F = \frac{W}{4}$. A set of head convolutional branches are then constructed on $\boldsymbol{f}$ to yield a class-wise center heatmap $\boldsymbol{P} \in \mathbb{R}^{H_F\times W_F\times N_{cls}}$ and task-specific prediction maps, such as 2D object size map and 3D object size map, etc. N_{cls} is the number of classes. CenterNet detects objects by their center points (peaks on $\boldsymbol{P}$) and the corresponding task-specific predictions from the peak positions.

Similar to [63], we build a baseline tracker by adding

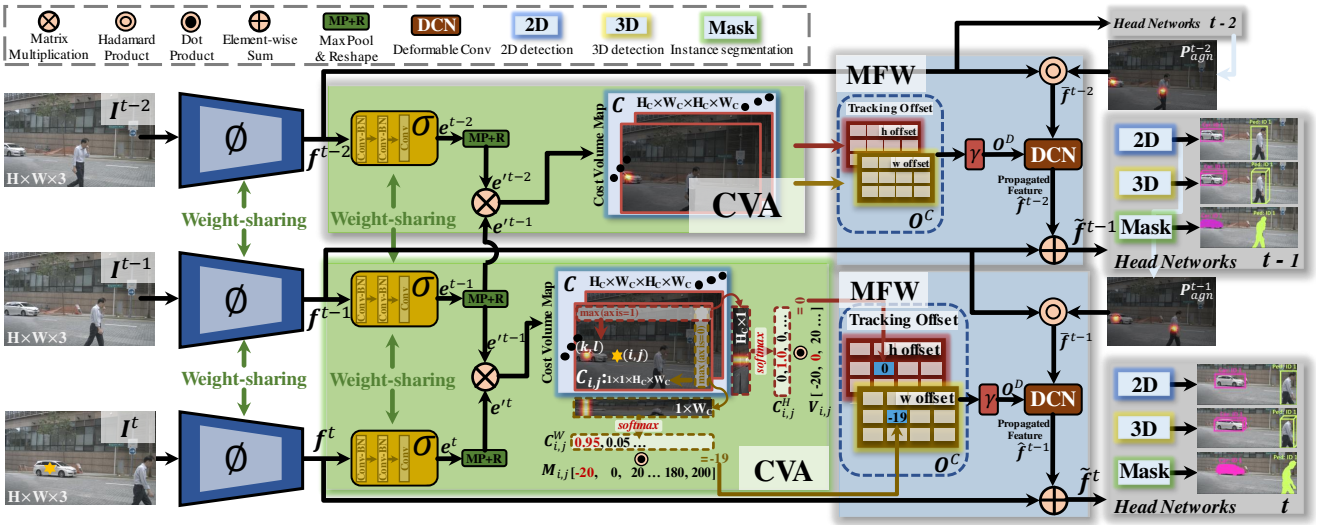

Figure 2. **Overview of TraDeS.** TraDeS may propagate features from multiple previous frames for object feature enhancement (*i.e.,* $T > 1$), which is not shown in the above figure for simplicity.

an extra head branch on CenterNet that predicts a tracking offset map $\boldsymbol{O}^B \in \mathbb{R}^{H_F \times W_F \times 2}$ for data association. $\boldsymbol{O}^B$ computes spatio-temporal displacements from all points at time t to the corresponding points at a previous time $t-\tau$.

4. TraDeS Tracker

Our Idea: Most previous joint detection and tracking methods perform a standalone detection without explicit input from tracking. In contrast, our aim is to integrate tracking cues into detection end-to-end, so as to improve detection for tough scenarios, which in return benefit tracking. To this end, we propose a Cost Volume based Association (CVA: § 4.1) module for learning re-ID embeddings and deriving object motions, and a Motion-guided Feature Warper (MFW: § 4.2) module for leveraging tracking cues from the CVA to propagate and enhance object features.

4.1. Cost Volume based Association

Cost Volume: Given two base features $\boldsymbol{f}^t$ and $\boldsymbol{f}^{t-\tau}$ from $\boldsymbol{I}^t$ and $\boldsymbol{I}^{t-\tau}$, we extract their re-ID embedding features by the embedding network $\sigma(\cdot)$, *i.e.,* $\boldsymbol{e}^t = \sigma(\boldsymbol{f}^t) \in \mathbb{R}^{H_F \times W_F \times 128}$, where $\sigma(\cdot)$ consists of three convolution layers. We utilize the extracted embeddings to construct a cost volume which stores dense matching similarities between one point and its corresponding point in two frames. To efficiently compute the cost volume, we first downsample the embeddings by a factor of 2 and obtain $\boldsymbol{e}' \in \mathbb{R}^{H_C \times W_C \times 128}$, where $H_C = \frac{H_F}{2}$ and $W_C = \frac{W_F}{2}$. Let us denote by $\boldsymbol{C} \in \mathbb{R}^{H_C \times W_C \times H_C \times W_C}$ the 4-dimensional cost volume for $\boldsymbol{I}^t$ and $\boldsymbol{I}^{t-\tau}$, which is computed by a single matrix multiplication of $\boldsymbol{e}'^t$ and $\boldsymbol{e}'^{t-\tau}$. Specifically, each element of $\boldsymbol{C}$ is calculated as:

$$C_{i,j,k,l} = \boldsymbol{e}'^t_{i,j} \boldsymbol{e}'^{t-\tau\top}_{k,l}, \tag{1}$$

where $C_{i,j,k,l}$ represents the embedding similarity between point (i,j) at time t and point (k,l) at time $t-\tau$. Here, a point refers to an entry on the feature map $\boldsymbol{f}$ or $\boldsymbol{e}'$.

Tracking Offset: Based on the cost volume $\boldsymbol{C}$, we calculate a tracking offset matrix $\boldsymbol{O} \in \mathbb{R}^{H_C \times W_C \times 2}$, which stores the spatio-temporal displacements for all points at time t to their corresponding points at time $t-\tau$. For illustration, we show the estimation procedure for $\boldsymbol{O}_{i,j} \in \mathbb{R}^2$ below.

As shown in Fig. 2, for an object x centered at point (i,j) at time t, we can fetch from $\boldsymbol{C}$ its corresponding two-dimensional cost volume map $\boldsymbol{C}_{i,j} \in \mathbb{R}^{H_C \times W_C}$. $\boldsymbol{C}_{i,j}$ stores the matching similarities among object x and all points at time $t-\tau$. Using $\boldsymbol{C}_{i,j}$, $\boldsymbol{O}_{i,j} \in \mathbb{R}^2$ is estimated by two steps: ***Step (i)*** $\boldsymbol{C}_{i,j}$ is first max pooled by $H_C \times 1$ and $1 \times W_C$ kernels, respectively, and then normalized by a softmax function[1], which results in $\boldsymbol{C}^W_{i,j} \in [0,1]^{1 \times W_C}$ and $\boldsymbol{C}^H_{i,j} \in [0,1]^{H_C \times 1}$. $\boldsymbol{C}^W_{i,j}$ and $\boldsymbol{C}^H_{i,j}$ consists of the likelihoods that object x appears on specified horizontal and vertical positions at time $t-\tau$, respectively. For example, $C^W_{i,j,l}$ is the likelihood that object x appears at the position $(*,l)$ at time $t-\tau$. ***Step (ii)*** Since $\boldsymbol{C}^W_{i,j}$ and $\boldsymbol{C}^H_{i,j}$ have provided the likelihoods that object x appears on specified positions at $t-\tau$. To obtain the final offsets, we predefine two offset templates for horizontal and vertical directions, respectively, indicating the actual offset values when x appears on those positions. Let $\boldsymbol{M}_{i,j} \in \mathbb{R}^{1 \times W_C}$ and $\boldsymbol{V}_{i,j} \in \mathbb{R}^{H_C \times 1}$ denote the horizontal and vertical offset templates for object x, respectively, which are computed by:

$$\begin{cases} M_{i,j,l} = (l-j) \times s & 1 \le l \le W_C \\ V_{i,j,k} = (k-i) \times s & 1 \le k \le H_C \end{cases}, \tag{2}$$

[1]We add a temperature of 5 into the softmax, such that the softmax output values are more discriminative.

where s is the feature stride of $\boldsymbol{e}'$ *w.r.t.* the input image, which is 8 in our case. $M_{i,j,l}$ refers to the horizontal offset when object x appears at the position $(*, l)$ at time $t-\tau$. The final tracking offset can be inferred by the dot product between the likelihoods and actual offset values as:

$$\boldsymbol{O}_{i,j} = [\boldsymbol{C}^{H\top}_{i,j}\boldsymbol{V}_{i,j}, \boldsymbol{C}^{W}_{i,j}\boldsymbol{M}^{\top}_{i,j}]^{\top}. \tag{3}$$

Because $\boldsymbol{O}$ is of $H_C \times W_C$, we upsample it with a factor of 2 and obtain $\boldsymbol{O}^C \in \mathbb{R}^{H_F \times W_F \times 2}$ that serves as motion cues for the MFW and is used for our data association.

Training: Since $\sigma(\cdot)$ is the only learnable part in the CVA module, the training objective of CVA is to learn an effective re-ID embedding $\boldsymbol{e}$. To supervise $\boldsymbol{e}$, we enforce the supervision on the cost volume rather than directly on $\boldsymbol{e}$ like other common re-ID losses. Let us first denote $Y_{ijkl} = 1$ when an object at location (i, j) at current time t appears at location (k, l) at previous time $t-\tau$; otherwise $Y_{ijkl} = 0$. Then, the training loss for CVA is calculated by the logistic regression in the form of the focal loss [22] as:

$$L_{CVA} = \frac{-1}{\sum_{ijkl} Y_{ijkl}} \sum_{ijkl} \begin{cases} \alpha_1 \log(C^{W}_{i,j,l}) \\ +\alpha_2 \log(C^{H}_{i,j,k}) & \text{if } Y_{ijkl} = 1 \\ 0 & \text{otherwise} \end{cases}, \tag{4}$$

where $\alpha_1 = (1 - C^{W}_{i,j,l})^{\beta}$ and $\alpha_2 = (1 - C^{H}_{i,j,k})^{\beta}$. β is the focal loss hyper-parameter. Since $C^{W}_{i,j,l}$ and $C^{H}_{i,j,k}$ are computed by softmax, they involve the embedding similarities not only between points (i, j) and (k, l) but also among point (i, j) and all other points in the previous frame. This is being said, while $C^{W}_{i,j,l}$ and $C^{H}_{i,j,k}$ being optimized to approach 1, it enforces an object to not only approach itself in the previous frame, but also *repel* other objects and background regions.

The CVA Characteristics: ***(i)*** Common re-ID loss only emphasizes intra-class variance, which may degrade detection performance. In contrast, our L_{CVA} in Eq. 4 not only emphasizes intra-class variance but also forces inter-class difference when learning embedding. We find such a treatment is more compatible with detection loss and learns effective embedding without hurting detection as evidenced in Tab. 1b. ***(ii)*** Because the tracking offset is predicted based on appearance embedding similarities, it can track objects under a wide range of motion and low frame rate as shown in Fig. 3 and Fig. 6, or even accurately track objects in different datasets with unseen large motion in training set as shown in Fig. 4. The predicted tracking offset can therefore serve as a robust motion cue to guide our feature propagation as in Tab. 1c. ***(iii)*** Compared to [45, 25] and CenterTrack [63] that only predict either embedding or tracking offset for data association, the CVA produces both embedding and tracking offset that are used for long-term data association (§ 4.3) and serve as motion cues for the MFW (§ 4.2).

4.2. Motion-guided Feature Warper

The MFW aims to take the predicted tracking offset $\boldsymbol{O}^C$ as motion clues to warp and propagate $\boldsymbol{f}^{t-\tau}$ to the current time so as to compensate and enhance $\boldsymbol{f}^t$. To achieve this goal, we perform an efficient temporal propagation via a single deformable convolution [12], which has been used for temporally aligning features in previous works [4, 3, 13]. Then, we enhance $\boldsymbol{f}^t$ by aggregating the propagated feature.

Temporal Propagation: To propagate feature maps, the deformable convolution (DCN) takes a spatio-temporal offset map and a previous feature as input and outputs a propagated feature, in which we estimate the input offset based on the $\boldsymbol{O}^C$ from the CVA module. Let us denote $\boldsymbol{O}^D \in \mathbb{R}^{H_F \times W_F \times 2K^2}$ as the input two-directional offset for DCN, where $K = 3$ is the kernel width or height of DCN. To generate $\boldsymbol{O}^D$, we pass $\boldsymbol{O}^C$ through a 3×3 convolution $\gamma(\cdot)$. We optionally incorporate the residual feature of $\boldsymbol{f}^t - \boldsymbol{f}^{t-\tau}$ as the input of $\gamma(\cdot)$ to provide more motion clues. Since our detection and segmentation are mainly based on object center features, instead of directly warping $\boldsymbol{f}^{t-\tau}$, we propagate a center attentive feature $\bar{\boldsymbol{f}}^{t-\tau} \in \mathbb{R}^{H_F \times W_F \times 64}$ from previous time. $\bar{\boldsymbol{f}}^{t-\tau}$ is computed as:

$$\bar{\boldsymbol{f}}^{t-\tau}_q = \boldsymbol{f}^{t-\tau}_q \circ \boldsymbol{P}^{t-\tau}_{agn}, \quad q = 1, 2, ..., 64, \tag{5}$$

where q is the channel index, $\circ$ is the Hadamard product, and $\boldsymbol{P}^{t-\tau}_{agn} \in \mathbb{R}^{H_F \times W_F \times 1}$ is the class agnostic center heatmap fetched from the $\boldsymbol{P}^{t-\tau}$ (as defined in § 3). Then, given $\boldsymbol{O}^D$ and $\bar{\boldsymbol{f}}^{t-\tau}$, the propagated feature is computed via a DCN as $\hat{\boldsymbol{f}}^{t-\tau} = DCN(\boldsymbol{O}^D, \bar{\boldsymbol{f}}^{t-\tau}) \in \mathbb{R}^{H_F \times W_F \times 64}$.

Feature Enhancement: When occlusion or motion blur occurs, objects could be missed by the detector. We propose to enhance $\boldsymbol{f}^t$ by aggregating the propagated feature $\hat{\boldsymbol{f}}^{t-\tau}$, on which the occluded and blurred objects may be visually legible. We denote the enhanced feature as $\tilde{\boldsymbol{f}}^{t-\tau}$, which is calculated by weighted summation as:

$$\tilde{\boldsymbol{f}}^t_q = \boldsymbol{w}^t \circ \boldsymbol{f}^t_q + \sum_{\tau=1}^{T} \boldsymbol{w}^{t-\tau} \circ \hat{\boldsymbol{f}}^{t-\tau}_q, \quad q = 1, 2, ..., 64, \tag{6}$$

where $\boldsymbol{w}^t \in \mathbb{R}^{H_F \times W_F \times 1}$ is the adaptive weight at time t and $\sum_{\tau=0}^{T} \boldsymbol{w}^{t-\tau}_{i,j} = 1$. T is the number of previous features used for aggregation. Similar to [24], $\boldsymbol{w}$ is predicted by two convolution layers followed by softmax function. We find that in experiment the weighted summation is slightly better than average summation. The enhanced feature $\tilde{\boldsymbol{f}}^t$ is then fed into the head networks to produce detection boxes and masks in the current frame. This can potentially recover missed objects and reduce false negatives, enabling complete tracklets and higher MOTA and IDF1 as in Tab. 1a.

4.3. Tracklet Generation

The overall architecture of TraDeS is shown in Fig. 2. Based on the enhanced feature $\tilde{\boldsymbol{f}}^t$, TraDeS produces 2D

and 3D boxes and instance masks by three different head networks. Afterwards, the generated detection and masks are connected to previous tracklets by our data association.

Head Networks: Each head network consists of several light-weight convolutions for yielding task-specific predictions. For 2D and 3D detection, we utilize the same head networks as in CenterNet [64]. For instance segmentation, we refer to the head network in CondInst [39], which is an instance segmentation method also based on center points.

Data Association: Given an enhanced detection or mask d centered at location (i, j), we perform a two-round data association as: ***DA-Round (i)*** We first associate it with the closest unmatched tracklet at time $t-1$ within the area centered at $(i,j)+\boldsymbol{O}^C_{i,j}$ with radius r, where r is the geometrical average of width and height of the detected box. Here, $\boldsymbol{O}^C_{i,j}$ only indicates the object tracking offsets between $\boldsymbol{I}^t$ and $\boldsymbol{I}^{t-1}$. ***DA-Round (ii)*** If d does not match any targets in the first round, we compute cosine similarities of its embedding $\boldsymbol{e}^t_{i,j}$ with all unmatched or history tracklet embeddings. d will be assigned to a tracklet if their similarity is the highest and larger than a threshold, *e.g.,* 0.3. *DA-Round (ii)* is capable of long-term associating. In case d fails to associate with any tacklets in the above two rounds, d starts a new tracklet.

TraDeS Loss: The overall loss function of TraDeS is defined as $L = L_{CVA} + L_{det} + L_{mask}$, where L_{det} is the 2D and 3D detection losses as in [64] and L_{mask} is the instance segmentation loss as in [39].

5. Experiments

5.1. Datasets and Implementation Details

MOT: We conduct 2D object tracking experiments on the MOT16 and MOT17 datasets [27], which have the same 7 training sequences and 7 test sequences but slightly different annotations. Frames are labeled at 25-30 FPS. For ablation study, we split the MOT17 training sequences into two halves and use one for training and the other for validation as in [63]. ***Metrics:*** We use common 2D MOT evaluation metrics [2]: Multiple-Object Tracking Accuracy (MOTA), ID F1 Score (IDF1), the number of False Negatives (FN), False Positives (FP), times a trajectory is Fragmented (Frag), Identity Switches (IDS), and the percentage of Mostly Tracked Trajectories (MT) and Mostly Lost Trajectories (ML).

nuScenes: We conduct 3D object tracking experiments on the newly released nuScenes [7], containing 7 classes, 700 training sequences, 150 validation sequences, and 150 test sequences. Videos are captured by 6 cameras of a moving car in a panoramic view and labeled at 2 FPS. Our TraDeS is a monocular tracker. ***Metrics:*** nuScenes designs more robust metrics, AMOTA and AMOTP, which are computed by weighted averages of MOTA and MOTP across score thresholds from 0 to 1. For fair comparison, we also report IDS_A that averages IDS in the same way.

MOTS: MOTS [41], an instance segmentation tracking dataset, is derived from the MOT dataset. MOTS has 4 training sequences and 4 test sequences. ***Metrics:*** The evaluation metrics are similar to those on MOT, which however are based on masks. Moreover, the MOTS adopts a Mask-based Soft Multi-Object Tracking Accuracy (sMOTSA).

YouTube-VIS: We also conduct instance segmentation tracking on YouTube-VIS [56], which contains 2,883 videos labeled at 6 FPS, 131K instance masks, and 40 object classes. ***Metrics:*** The YouTube-VIS adopts a mask tracklets based average precision (AP) for evaluation.

Compared to MOT and MOTS, nuScenes and YouTube-VIS are of low frame rate and large motion, because only key frames are labeled and cameras are moving. In our experiments, only labeled frames are used as input.

Implementation Details: We adopt the same experimental settings as CenterTrack [63], such as backbone, image sizes, pretraining, score thresholds, etc. Specifically, we adopt the DLA-34 [60] as the backbone network $\phi(\cdot)$. Our method is optimized with 32 batches and learning rate (lr) $1.25e-4$ dropping by a factor of 10. For MOT and MOTS, TraDeS is trained for 70 epochs where lr drops at epoch 60 with image size of 544×960. For nuScenes, TraDeS is trained for 35 epochs where lr drops at epoch 30 with image size of 448×800. For YouTube-VIS, TraDeS is first pretrained on COCO instance segmentation [23] following the static image training scheme in [63] and then finetuned on YouTube-VIS for 16 epochs where lr drops at epoch 9. Image size is of 352×640. We test the runtime on a 2080Ti GPU. In Eq. 6, we set $T = 2$ by default for MOT and MOTS. We set $T = 1$ for nuScenes and YouTube-VIS due to their low frame rate characteristic mentioned above. In training, we randomly select T frames out of nearby R_t frames, where R_t is 10 for MOT and MOTS and 5 for nuScenes and YouTube-VIS. During inference, only previous T consecutive frames are used. Ablation experiments are conducted on the MOT17 dataset. In ablations, all variants without the CVA module perform the *DA-Round (i)* by predicting a tracking offset $\boldsymbol{O}^B$ as in the baseline tracker (§ 3).

5.2. Ablation Studies

Effectiveness of TraDeS: As shown in Tab. 1a, we compare our proposed CVA (§ 4.1), MFW (§ 4.2), and TraDeS (§ 4) with our baseline tracker (§ 3) and CenterTrack [63]. ***CVA:*** Compared to the baseline, the CVA achieves *better tracking* by reducing 60% IDS and improving 7.2 IDF1, validating the effect of our tracking offset, re-ID embedding, and the two-round data association. ***MFW:*** For ablation, we directly add the MFW to the baseline tracker. Since the tracking offset $\boldsymbol{O}^C$ is unavailable in the baseline, we only use $\boldsymbol{f}^t - \boldsymbol{f}^{t-\tau}$

Scheme	MOTA↑	IDF1↑	IDS↓	FN↓	FP↓
CenterTrack[63]	66.1	64.2	528	28.4%	4.5%
Baseline	64.8	59.5	1055	31.0%	2.3 %
Baseline+CVA	66.5	66.7	415	30.6%	**2.2%**
Baseline+MFW	66.3	65.7	606	29.5%	3.0%
TraDeS	**68.2**	**71.7**	**285**	**27.8%**	3.5%

(a) **Effectiveness of each proposed module**: we evaluate the proposed CVA (§ 4.1), MFW (§ 4.2), and overall TraDeS (§ 4). "Baseline+CVA+MFW" is represented by "TraDeS".

Scheme		MOTA↑	IDF1↑	IDS↓	FN↓	FP↓
	Baseline	64.8	59.5	1055	31.0%	2.3%
w/o ***DA-Round (ii)***	+CE embedding	63.7	59.6	1099	32.1%	**2.2%**
	+CVA	65.5	60.9	936	**30.6%**	**2.2%**
w/ ***DA-Round (ii)***	+CE embedding	64.5	64.3	671	32.1%	**2.2%**
	+CVA	**66.5**	**66.7**	**415**	**30.6%**	**2.2%**

(b) **CVA *vs.* Common embedding**: Common embedding loss $L_{CEembed}$ may downgrade detection performance, while our CVA learns an effective embedding without hurting detection. As "Baseline" does not have embedding, it only performs *DA-Round(i)*.

Scheme		MOTA↑	IDF1↑	IDS↓	FN↓	FP↓
	Baseline+CVA	66.5	66.7	415	30.6%	**2.2%**
TraDeS w/	$\boldsymbol{f}^t - \boldsymbol{f}^{t-\tau}$ only	67.1	68.8	**273**	29.9%	2.5%
	$\boldsymbol{f}^t - \boldsymbol{f}^{t-\tau}$ & $\boldsymbol{O}^C$	**68.2**	**71.7**	285	**27.8%**	3.5%

(c) **Motion cues:** In MFW, we evaluate different motion cues as the input of $\gamma(\cdot)$ to predict the DCN input offset $\boldsymbol{O}^D$. Ablations are based on baseline with CVA.

Scheme	MOTA↑	IDF1↑	IDS↓	FN↓	FP↓	Time(ms)↓
$T = 1$	67.8	69.0	350	28.2%	3.4%	46
$T = 2$	**68.2**	**71.7**	285	**27.8%**	3.5%	57
$T = 3$	67.5	69.9	**283**	29.2%	**2.8%**	70

(d) **Number of previous features**: We evaluate the MFW when aggregating different numbers of previous features.

Table 1. **Ablation studies** on the MOT17 validation set. MOTA and IDF1 reflect the comprehensive tracking performance, while FN and FP reflect the detection performance. Lower FN means more missed objects are recovered. ↓ denotes lower is better. ↑ denotes higher is better.

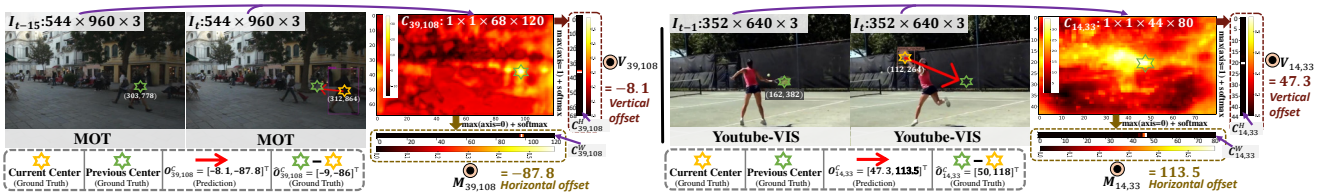

Figure 3. **CVA workflow visualization:** the cost volume map $\boldsymbol{C}$ and tracking offset $\boldsymbol{O}^C$ under low frame rate (left) and large motion (right).

as motion cues to predict the DCN offset $\boldsymbol{O}^D$. Compared to the baseline, the MFW achieves *better detection* by reducing 1.5% FN, *i.e.,* recovering more missed objects, though FP is slightly increased. Moreover, we observe that the MFW also reduces 43% IDS and improves 6.2 IDF1. It validates that *detection is the cornerstone for tracking performance*, where improved detection can yield more stable and consistent tracklets. ***TraDeS:*** With the help of CVA, TraDeS reduces IDS from 606 to 285. Moreover, in TraDeS, the robust tracking offsets $\boldsymbol{O}^C$ from CVA guides the feature propagation in MFW, which significantly decreases FN from 29.5% to 27.8%. Better IDS and missed object recovery (↓FN) together improve our comprehensive tracking performance, achieving 68.2 MOTA and 71.7 IDF1. TraDeS also achieves better results than the recent *JDT* method CenterTrack [63].

Effectiveness of the CVA Module: We study the two major characteristics of the proposed CVA module as mentioned in § 4.1. ***(i):*** First, we add the re-ID embedding network $\sigma(\cdot)$ into the baseline tracker, which is supervised by a common re-ID loss, *e.g.,* the cross-entropy loss $L_{CEembed}$ as in [45, 61]. We denote the learned embedding as CE embedding, which is used to perform our two-round data association. As shown in Tab. 1b, with *DA-Round (ii)*, CE embedding helps baseline improve IDF1 and reduce IDS, as long-term data association is enabled by using the re-ID embedding to match history tracklets. However, we observe that CE embedding cannot improve MOTA as detection performance is degraded (+1.1% FN). Next, we still add $\sigma(\cdot)$ into the baseline tracker, which however is supervised by our CVA module. Tab. 1b shows that our CVA module not only learns an effective re-ID embedding as CE embedding but also slightly improves detection performance, which clearly leads to a higher MOTA. We argue that this is because common re-ID loss only emphasizes intra-class variance, which may not be compatible with detection loss in joint training as indicted in [9]. In contrast, our proposed L_{CVA} in Eq. 4 supervises the re-ID embedding via the cost volume and considers both intra-class and inter-class difference. ***(ii):*** We visualize the predicted cost volume map $\boldsymbol{C}$ and tracking offset $\boldsymbol{O}^C$ in Fig. 3. The CVA accurately predicts the tracking offset for an object under low frame rate or large motion. Moreover, $\boldsymbol{O}^C$ even accurately tracks objects in a new dataset with unseen large motion in training as shown in Fig. 4. Visualization of $\boldsymbol{O}^C$ on more samples are shown in Fig. 6. These examples indicate the CVA is able to predict tracking offsets for objects with a wide range of motion and provide robust motion cues.

Effectiveness of the MFW Module: ***DCN:*** In Tab. 1c, we use different motion clues to predict the DCN input offset $\boldsymbol{O}^D$. We find that the tracking offset $\boldsymbol{O}^C$ is the key to reduce FN and recover more missed objects. It validates that the proposed $\boldsymbol{O}^C$ is a robust tracking cue for guiding feature propagation and assisting detection. Moreover, we visualize the predicted $\boldsymbol{O}^D$ in Fig. 5. The DCN successfully samples the center features at the previous frames even if the car in the middle image has dramatic displacements. ***Number of Previous Features:*** As in Eq. 6, the MFW aggregates the current feature with T previous features. We evaluate the MFW with different T as shown in Tab. 1d, and find that we

Method	Publication	Year	Joint	MOTA↑	IDF1↑	MT↑	ML↓	FP↓	FN↓	Frag↓	IDS↓	Time(ms)↓
MOT16 Test Set												
SORT[5]^{Online}	ICIP	2016		59.8	53.8	25.4%	22.7%	8,698	63,245	1,835	1,423	17+D
MCMOT-HDM[21]^{Offline}	ECCV	2016		62.4	51.6	31.5%	24.2%	9,855	57,257	1,318	1,394	27+D
POI[59]^{Online}	ECCVW	2016		66.1	65.1	34.0%	20.8%	5,061	55,914	3,093	805	101+D
DeepSORT[49]^{Online}	ICIP	2017		61.4	62.2	32.8%	18.2%	12,852	56,668	2,008	781	25+D
VMaxx[42]^{Online}	ICIP	2018		62.6	49.2	32.7%	21.1%	10,604	56,182	1,534	1,389	154+D
RAN[14]^{Online}	WACV	2018		63.0	63.8	39.9%	22.1%	13,663	53,248	1,251	482	625+D
TAP[65]^{Online}	ICPR	2018		64.8	73.5	38.5%	21.6%	12,980	50,635	1,048	571	55+D
TubeTK[28]^{Offline}	CVPR	2020	✓	64.0	**59.4**	33.5%	**19.4 %**	10,962	53,626	**1,366**	**1,117**	1000
JDE[45]^{Online}	ECCV	2020	✓	64.4	55.8	**35.4%**	**20.0%**	-	-	-	1,544	45
CTracker^{Online}	ECCV	2020	✓	**67.6**	57.2	32.9%	23.1%	**8,934**	**48,305**	3,112	1,897	29
TraDeS (Ours)^{Online}	CVPR	2021	✓	**70.1**	**64.7**	**37.3 %**	**20.0 %**	**8,091**	**45,210**	**1,575**	**1,144**	57
MOT17 Test Set												
CenterTrack*[63]^{Online}	ECCV	2020	✓	67.8	64.7	34.6%	24.6%	18,498	160,332	6,102	3,039	57
TraDeS* (Ours)^{Online}	CVPR	2021	✓	68.9	67.2	35.0%	22.7%	19,701	152,622	6,033	3,147	57
DAN[37]^{Online}	TPAMI	2019		52.4	49.5	21.4%	30.7%	25,423	234,592	14,797	8,431	159+D
Tracktor+CTdet[1]^{Online}	ICCV	2019		54.4	56.1	25.7%	29.8%	44,109	210,774	-	2,574	-
TubeTK[28]^{Offline}	CVPR	2020	✓	63.0	58.6	31.2%	**19.9%**	27,060	177,483	**5,727**	4,137	333
CTracker[29]^{Online}	ECCV	2020	✓	66.6	57.4	32.2%	24.2%	**22,284**	160,491	9,114	5,529	29
CenterTrack[63]^{Online}	ECCV	2020	✓	**67.3**	**59.9**	**34.9%**	24.8%	23,031	**158,676**	-	**2,898**	57
TraDeS (Ours)^{Online}	CVPR	2021	✓	**69.1**	**63.9**	**36.4 %**	**21.5 %**	**20,892**	**150,060**	**4,833**	**3,555**	57

Table 2. **Results of 2D object tracking on the MOT test set under the private detection protocol.** "Joint" indicates joint detection and tracking in a single model, *i.e.,* no external detections. "⋆" indicates that Track Re-birth [63] is used. The top two results in the "Joint" manner without Track Re-birth are highlighted in **red** and **blue**, respectively. +D indicates the additional detection time [31].

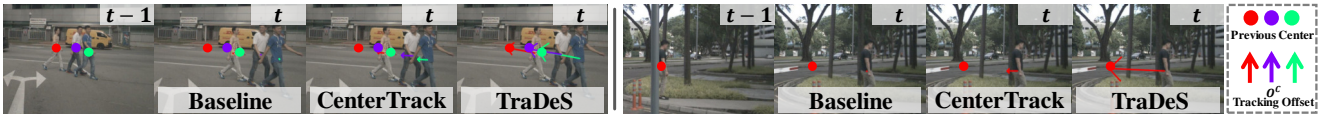

Figure 4. **Visualized O^C on nuScenes.** All models are only trained on MOT but tested on nuScenes, where nuScenes has much larger object motions than MOT. TraDeS successfully tracks objects with unseen large motion in training dataset, but baseline and CenterTrack fail.

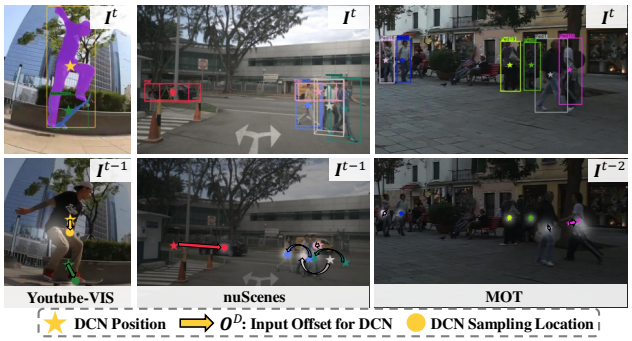

Figure 5. **Visualization of DCN input offset O^D.** The DCN kernel at ★ is translated by ⇨ and samples the previous feature at ●. For clear visualization, we only show the sampling center of the DCN kernel as depicted by ● in $I^{t-\tau}$. The previous image is highlighted by the previous class agnostic heatmap $P_{agn}^{t-\tau}$.

achieve the best speed-accuracy trade-off when $T = 2$.

5.3. Benchmark Evaluations

MOT: As shown in Tab. 2, we compare the proposed TraDeS tracker with the state-of-the-art 2D trackers on the MOT16 and MOT17 test sets. Our TraDeS tracker outperforms the second best tracker by *2.5* MOTA and *1.8* MOTA on MOT16 and MOT17, respectively, running at *15 FPS*. Compared to joint detection and tracking algorithms, we achieve the best results on most metrics, *e.g.,* MOTA, IDF1, MT, FN, etc.

nuScenes: As shown in Tab. 3, we compare TraDeS with the state-of-the-art monocular 3D trackers on nuScenes. There exists extreme class imbalance in nuScenes dataset, *e.g.,* car and pedestrian has over 82% data. Since class imbalance is not our focus, we mainly evaluate on major classes: car and pedestrian. Tab. 3 shows that the TraDeS tracker outperforms other monocular trackers by a large margin on all metrics.

MOTS: As shown in Tab. 4, we compare TraDeS with the recent instance segmentation tracker TrackR-CNN on the MOTS test set. TrackR-CNN is based on Mask R-CNN [17] and also temporally enhances object features. The TraDeS tracker outperforms TrackR-CNN by a large margin in terms of both accuracy and speed.

YouTube-VIS: As shown in Tab. 5, TraDeS notably improves AP by 6.2 over the baseline. TraDeS achieves compet-

Classes	Car (57% 58,317GTs)			Pedestrian (25% 25,423GTs)			All (100% 101,897GTs)			
nuScenes Val set	AMOTA↑	AMOTP↓	IDS$_A$ ↓	AMOTA↑	AMOTP↓	IDS$_A$ ↓	AMOTA↑	AMOTP↓	IDS$_A$ ↓	Time
Our Baseline	11.1	1.39	6,985	0.0	1.73	4,336	4.3	1.65	1,792	37ms
CenterTrack[63]	26.1	1.11	3,217	5.9	1.50	1,970	6.8	1.54	813	45ms
TraDeS (Ours)	**29.6**	**0.98**	**3,035**	**10.6**	**1.42**	**1,434**	**11.8**	**1.48**	**699**	39ms
Classes	Car (57% 68,518GTs)			Pedestrian (28% 34,010GTs)			All (100% 119,565GTs)			
nuScenes Test set	AMOTA↑	AMOTP↓	IDS$_A$ ↓	AMOTA↑	AMOTP↓	IDS$_A$ ↓	AMOTA↑	AMOTP↓	IDS$_A$ ↓	Time
Our Baseline	6.2	1.47	9,450	0.0	1.70	5,191	1.0	1.66	2,252	37ms
Mapillary[34]+AB3D[47]	12.5	1.61	-	0.0	1.87	-	1.8	1.80	-	-
PointPillars[20]+AB3D[47]	9.4	1.40	-	3.9	1.68	-	2.9	1.70	-	-
CenterTrack[63]	20.2	1.19	-	3.0	1.50	-	4.6	1.54	-	45ms
TraDeS (Ours)	**23.2**	**1.07**	**4,293**	**9.9**	**1.38**	**1,979**	**5.9**	**1.49**	**964**	39ms

Table 3. **Results of 3D object tracking on the nuScenes dataset.** We compare with the state-of-the-art monocular 3D tracking methods. We mainly assess the major classes: car and pedestrian. We also list "All" for reference, which is the average among all the 7 classes.

Method	Publication	Year	sMOTSA ↑	IDF1 ↑	MOTSA ↑	MOTSP ↑	MODSA ↑	MT↑	ML↓	FP↓	FN↓	IDS ↓	Time
TrackR-CNN [41]	CVPR	2019	40.6	42.4	55.2	76.1	56.9	38.7%	21.6%	**1,261**	12,641	567	500ms
TraDeS (Ours)	CVPR	2021	**50.8**	**58.7**	**65.5**	**79.5**	**67.0**	**49.4%**	**18.3%**	1,474	**9,169**	**492**	87ms

Table 4. **Results of instance segmentation tracking on the MOTS test set.**

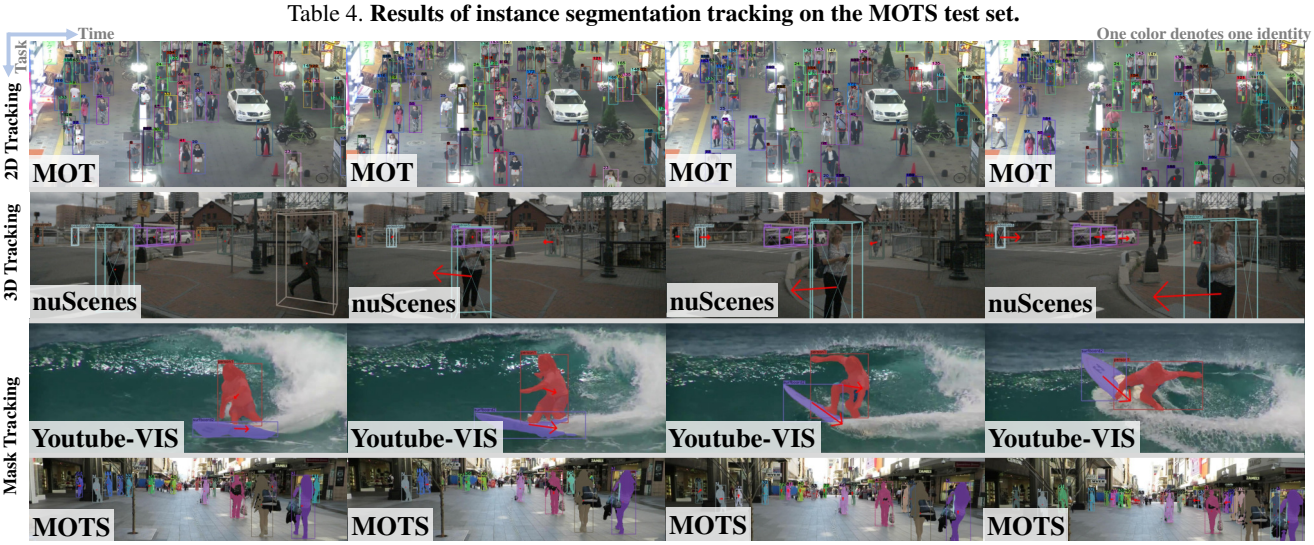

Figure 6. **Visualization that TraDeS tracks objects on three tasks.** Red arrow is the tracking offset $\boldsymbol{O}^C$ *w.r.t.* the previous frame $\boldsymbol{I}^{t-1}$.

Method	Publication	AP	AP$_{50}$	AP$_{75}$
OSMN(mask propagation)[57]	CVPR'18	23.4	36.5	25.7
FEELVOS[40]	CVPR'19	26.9	42.0	29.7
OSMN(track-by-detect)[57]	CVPR'18	27.5	45.1	29.1
MaskTrack R-CNN[56]	ICCV'19	30.3	51.1	32.6
SipMask[8]	ECCV'20	32.5	**53.0**	**33.3**
Our Baseline		26.4	43.2	26.8
TraDeS (Ours)	CVPR'21	**32.6**	52.6	32.8

Table 5. **Results of instance segmentation tracking on the YouTube-VIS validation set.**

itive performance compared to other state-of-the-art instance segmentation trackers. We observe that TraDeS outperforms the baseline tracker by a large margin on both nuScenes and YouTube-VIS. We argue that this is because the baseline cannot well predict the tracking offset $\boldsymbol{O}^B$ with a single image in case these datasets are of low frame rate and large motion.

6. Conclusion

This work presents a novel online joint detection and tracking model, TraDeS, focusing on exploiting tracking cues to help detection and in return benefit tracking. TraDeS is equipped with two proposed modules, CVA and MFW. The CVA learns a dedicatedly designed re-ID embedding and models object motions via a 4d cost volume. While the MFW takes the motions from CVA as the cues to propagate previous object features to enhance the current detection or segmentation. Exhaustive experiments and ablations on 2D tracking, 3D tracking and instance segmentation tracking validate both effectiveness and superiority of our approach.

Acknowledgement. This work is supported in part by a gift grant from Horizon Robotics and National Science Foundation Grant CNS1951952. We thank Sijia Chen and Li Huang from Horizon Robotics for helpful discussion.

References

[1] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *ICCV*, 2019. 1, 2, 7

[2] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008. 5

[3] Gedas Bertasius, Christoph Feichtenhofer, Du Tran, Jianbo Shi, and Lorenzo Torresani. Learning temporal pose estimation from sparsely-labeled videos. In *NeurIPS*, 2019. 4

[4] Gedas Bertasius, Lorenzo Torresani, and Jianbo Shi. Object detection in video with spatiotemporal sampling networks. In *ECCV*, 2018. 4

[5] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *ICIP*, 2016. 1, 2, 7

[6] Guillem Brasó and Laura Leal-Taixé. Learning a neural solver for multiple object tracking. In *CVPR*, 2020. 2

[7] 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 *CVPR*, 2020. 5

[8] Jiale Cao, Rao Muhammad Anwer, Hisham Cholakkal, Fahad Shahbaz Khan, Yanwei Pang, and Ling Shao. Sipmask: Spatial information preservation for fast image and video instance segmentation. *ECCV*, 2020. 8

[9] Di Chen, Shanshan Zhang, Wanli Ouyang, Jian Yang, and Ying Tai. Person search via a mask-guided two-stream cnn model. In *ECCV*, 2018. 1, 6

[10] Qifeng Chen and Vladlen Koltun. Full flow: Optical flow estimation by global optimization over regular grids. In *CVPR*, 2016. 2

[11] Robert T Collins. A space-sweep approach to true multi-image matching. In *CVPR*, 1996. 2

[12] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017. 4

[13] Jiajun Deng, Yingwei Pan, Ting Yao, Wengang Zhou, Houqiang Li, and Tao Mei. Single shot video object detector. *TMM*, 2020. 4

[14] Kuan Fang, Yu Xiang, Xiaocheng Li, and Silvio Savarese. Recurrent autoregressive networks for online multi-object tracking. In *WACV*, 2018. 7

[15] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Detect to track and track to detect. In *ICCV*, 2017. 2

[16] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 2009. 2

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 7

[18] Sunghoon Im, Hae-Gon Jeon, Stephen Lin, and In So Kweon. Dpsnet: End-to-end deep plane sweep stereo. 2019. 2

[19] Chanho Kim, Fuxin Li, Arridhana Ciptadi, and James M Rehg. Multiple hypothesis tracking revisited. In *ICCV*, 2015. 2

[20] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, 2019. 8

[21] Byungjae Lee, Enkhbayar Erdenee, Songguo Jin, Mi Young Nam, Young Giu Jung, and Phill Kyu Rhee. Multi-class multi-object tracking using changing point detection. In *ECCV*, 2016. 7

[22] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 4

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 5

[24] Songtao Liu, Di Huang, and Yunhong Wang. Learning spatial fusion for single-shot object detection. *arXiv preprint arXiv:1911.09516*, 2019. 4

[25] Zhichao Lu, Vivek Rathod, Ronny Votel, and Jonathan Huang. Retinatrack: Online single stage joint detection and tracking. In *CVPR*, 2020. 1, 2, 4

[26] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. *arXiv preprint arXiv:2101.02702*, 2021. 1, 2

[27] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*. 1, 5

[28] Bo Pang, Yizhuo Li, Yifan Zhang, Muchen Li, and Cewu Lu. Tubetk: Adopting tubes to track multi-object in a one-step training model. In *CVPR*, 2020. 2, 7

[29] Jinlong Peng, Changan Wang, Fangbin Wan, Yang Wu, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking. In *ECCV*, 2020. 1, 2, 7

[30] Lorenzo Porzi, Markus Hofinger, Idoia Ruiz, Joan Serrat, Samuel Rota Bulo, and Peter Kontschieder. Learning multi-object tracking and segmentation from automatic annotations. In *CVPR*, 2020. 1

[31] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 2, 7

[32] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 1, 2

[33] Samuel Schulter, Paul Vernaza, Wongun Choi, and Manmohan Chandraker. Deep network flow for multi-object tracking. In *CVPR*, 2017. 2

[34] Andrea Simonelli, Samuel Rota Rota Bulò, Lorenzo Porzi, Manuel López-Antequera, and Peter Kontschieder. Disentangling monocular 3d object detection. In *ICCV*, 2019. 8

[35] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018. 2

[36] Peize Sun, Yi Jiang, Rufeng Zhang, Enze Xie, Jinkun Cao, Xinting Hu, Tao Kong, Zehuan Yuan, Changhu Wang, and Ping Luo. Transtrack: Multiple-object tracking with transformer. *arXiv preprint arXiv:2012.15460*, 2020. 2

[37] Shijie Sun, Naveed Akhtar, HuanSheng Song, Ajmal S Mian, and Mubarak Shah. Deep affinity network for multiple object tracking. *IEEE TPAMI*, 2019. 7

[38] Siyu Tang, Mykhaylo Andriluka, Bjoern Andres, and Bernt Schiele. Multiple people tracking by lifted multicut and person re-identification. In *CVPR*, 2017. 1, 2

[39] Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. In *ECCV*, 2020. 5

[40] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen. Feelvos: Fast end-to-end embedding learning for video object segmentation. In *CVPR*, 2019. 8

[41] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. Mots: Multi-object tracking and segmentation. In *CVPR*, 2019. 5, 8

[42] Xingyu Wan, Jinjun Wang, Zhifeng Kong, Qing Zhao, and Shunming Deng. Multi-object tracking using online metric learning with long short-term memory. In *ICIP*, 2018. 7

[43] Manchen Wang, Joseph Tighe, and Davide Modolo. Combining detection and tracking for human pose estimation in videos. In *CVPR*, 2020. 1

[44] Yongxin Wang, Kris Kitani, and Xinshuo Weng. Joint object detection and multi-object tracking with graph neural networks. *arXiv preprint arXiv:2006.13164*, 2020. 1

[45] Zhongdao Wang, Liang Zheng, Yixuan Liu, and Shengjin Wang. Towards real-time multi-object tracking. In *ECCV*, 2020. 1, 2, 4, 6, 7

[46] Longyin Wen, Wenbo Li, Junjie Yan, Zhen Lei, Dong Yi, and Stan Z Li. Multiple target tracking based on undirected hierarchical relation hypergraph. In *CVPR*, 2014. 2

[47] Xinshuo Weng and Kris Kitani. 3d multi-object tracking: A baseline and new evaluation metrics. In *IROS*, 2020. 8

[48] 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 *CVPR*, 2020. 2

[49] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, 2017. 1, 2, 7

[50] Jialian Wu, Chunluan Zhou, Ming Yang, Qian Zhang, Yuan Li, and Junsong Yuan. Temporal-context enhanced detection of heavily occluded pedestrians. In *CVPR*, 2020. 1

[51] Tong Xiao, Shuang Li, Bochao Wang, Liang Lin, and Xiaogang Wang. Joint detection and identification feature learning for person search. In *CVPR*, 2017. 1, 2

[52] Jiarui Xu, Yue Cao, Zheng Zhang, and Han Hu. Spatial-temporal relation networks for multi-object tracking. In *ICCV*, 2019. 1, 2

[53] Jia Xu, René Ranftl, and Vladlen Koltun. Accurate optical flow via direct cost volume processing. In *CVPR*, 2017. 2

[54] Zhenbo Xu, Wei Zhang, Xiao Tan, Wei Yang, Huan Huang, Shilei Wen, Errui Ding, and Liusheng Huang. Segment as points for efficient online multi-object tracking and segmentation. In *ECCV*, 2020. 2

[55] Jiayu Yang, Wei Mao, Jose M Alvarez, and Miaomiao Liu. Cost volume pyramid based depth inference for multi-view stereo. In *CVPR*, 2020. 2

[56] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *ICCV*, 2019. 5, 8

[57] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K. Katsaggelos. Efficient video object segmentation via network modulation. In *CVPR*, 2018. 8

[58] Junbo Yin, Wenguan Wang, Qinghao Meng, Ruigang Yang, and Jianbing Shen. A unified object motion and affinity model for online multi-object tracking. In *CVPR*, 2020. 2

[59] Fengwei Yu, Wenbo Li, Quanquan Li, Yu Liu, Xiaohua Shi, and Junjie Yan. Poi: Multiple object tracking with high performance detection and appearance feature. In *ECCV Workshops*, 2016. 7

[60] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *CVPR*, 2018. 5

[61] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *arXiv preprint arXiv:2004.01888*, 2020. 2, 6

[62] Zheng Zhang, Dazhi Cheng, Xizhou Zhu, Stephen Lin, and Jifeng Dai. Integrated object detection and tracking with tracklet-conditioned detection. *arXiv preprint arXiv:1811.11167*, 2018. 2

[63] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *ECCV*, 2020. 1, 2, 4, 5, 6, 7, 8

[64] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 1, 2, 5

[65] Zongwei Zhou, Junliang Xing, Mengdan Zhang, and Weiming Hu. Online multi-target tracking with tensor-based high-order graph matching. In *ICPR*, 2018. 7

[66] Ji Zhu, Hua Yang, Nian Liu, Minyoung Kim, Wenjun Zhang, and Ming-Hsuan Yang. Online multi-object tracking with dual matching attention networks. In *ECCV*, 2018. 2