OffsetNet: Towards Efficient Multiple Object Tracking, Detection, and Segmentation

Wei Zhang, Jiaming Li, Meng Xia, Xu Gao, Xiao Tan, Yifeng Shi, Zhenhua Huang, and Guanbin Li, *Member, IEEE*

***Abstract*—Offset-based representation has emerged as a promising approach for modeling semantic relations between pixels and object motion, demonstrating efficacy across various computer vision tasks. In this paper, we introduce a novel one-stage multi-tasking network tailored to extend the offset-based approach to MOTS. Our proposed framework, named OffsetNet, is designed to concurrently address amodal bounding box detection, instance segmentation, and tracking. It achieves this by formulating these three tasks within a unified pixel-offset-based representation, thereby achieving excellent efficiency and encouraging mutual collaborations. OffsetNet achieves several remarkable properties: first, the encoder is empowered by a novel Memory Enhanced Linear Self-Attention (MELSA) block to efficiently aggregate spatial-temporal features; second, all tasks are decoupled fairly using three lightweight decoders that operate in a one-shot manner; third, a novel cross-frame offsets prediction module is proposed to enhance the robustness of tracking against occlusions. With these merits, OffsetNet achieves 76.83% HOTA on KITTI MOTS benchmark, which is the best result without relying on 3D detection. Furthermore, OffsetNet achieves 74.83% HOTA at 50 FPS on the KITTI MOT benchmark, which is nearly 3.3 times faster than CenterTrack with better performance. We hope our approach will serve as a solid baseline and encourage future research in this field.**

***Index Terms*—Multi-Object tracking, object detection, object segmentation.**

I. Introduction

MULTI-OBJECT tracking and segmentation (MOTS) which jointly performs pixel-level instance discrimination and object tracking in a dynamic scene, attracts increasing attention recently [1], [2], [3], [4], [5]. Since the multi-tasking nature of MOTS is well-suited to practical applications such as autonomous driving, video surveillance and video analysis, exploring efficient and all-in-one networks is important to this field.

Though great efforts have been made, developing an efficient multi-tasking MOTS network still imposes great challenges. Recent two-stage methods [1], [2], [6] usually treat instance segmentation and multi-object tracking as two successive stages independently, resulting in heavy network architectures without end-to-end optimization. On the other hand, the advanced one-stage offline method STEm-Seg [4] requires the entire video clip as input, thus limiting the usage in on-line scenarios. CCPNet [7] proposes the first one-stage and online MOTS which replaces the secondary ReID feature extraction sub-network by a simple spatial max-pooling operation. However, since the ReID feature extraction in CCPNet still relies heavily on the results of instance segmentation, it is hard to be optimized in an end-to-end scheme, thus is sub-optimal in tracking performance. To date, none of the existing MOTS methods consider all the sub-tasks in a unified manner, leading to significant limitations in both efficiency and multi-task performance.

Recently, offset-based representation which efficiently models semantic relations and motions of objects using pixel-level displacements has been proved successful in various vision tasks. Such as CenterTrack [8] proposes to formulate tracking as predicting inter-frame offsets between instances and achieves

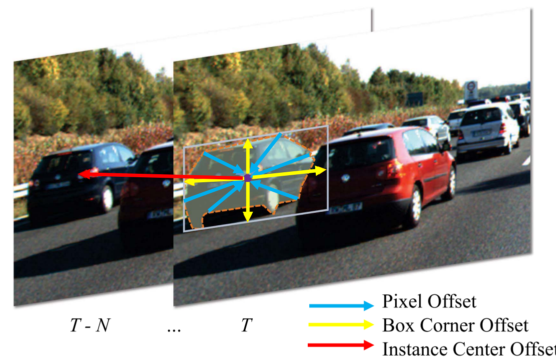

Fig. 1. This figure intuitively illustrates the basic concepts of our proposed offset-based MOTS method. The arrow lines with different colors represent different types of offsets produced by our proposed *OffsetNet*, which simultaneously addresses amodal bounding box detection (as shown by the box with solid lines), instance segmentation (as shown by the mask with dotted lines) and tracking over multiple frames.

Received 3 March 2023; revised 9 September 2024; accepted 10 October 2024. Date of publication 4 November 2024; date of current version 9 January 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62322608, in part by the Shenzhen Science and Technology Program under Grant JCYJ20220530141211024, and in part by GuangDong Basic and Applied Basic Research Fund under Grant 2024A1515010255. Recommended for acceptance by M. Leordeanu. *(Corresponding author: Guanbin Li.)*

Wei Zhang, Xu Gao, Xiao Tan, and Yifeng Shi are with Baidu Inc, Beijing 100085, China (e-mail: zhangwei99@baidu.com; gaoxu03@baidu.com; tanxiao01@baidu.com; shiyifeng@baidu.com).

Jiaming Li and Meng Xia are with the School of Computer Science and Engineering, Research Institute of Sun Yat-sen University in Shenzhen, Sun Yat-sen University, Guangzhou 510006, China (e-mail: lijm48@mail2.sysu.edu.cn; xiam9@mail2.sysu.edu.cn).

Zhenhua Huang is with the School of Computer Science, South China Normal University, Guangzhou 510631, China (e-mail: huangzhenhua@m.scnu.edu.cn).

Guanbin Li is with the School of Computer Science and Engineering, Research Institute of Sun Yat-sen University in Shenzhen, Sun Yat-sen University, Guangzhou 510006, China, and also with Peng Cheng Laboratory, Shenzhen 518066, China (e-mail: liguanbin@mail.sysu.edu.cn).

Digital Object Identifier 10.1109/TPAMI.2024.3485644

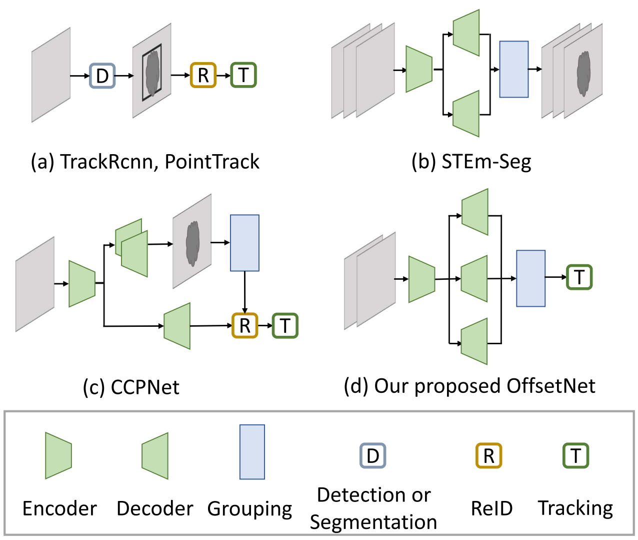

Fig. 2. *Comparison of different MOTS frameworks.* (a) Standard two-stage methods which follow tracking-by-detection [1] or tracking-by-segmentation [2] paradigm; (b) One-stage offline method which takes one video clip as input and groups pixels belonging to a specific object instance over an entire video clip [4]; (c) CCPNet [7] provides the first one-stage and online method while Reid still rely heavily on the segmentation results. Bounding box detection is not considered in CCPNet; (d) Our proposed *OffsetNet* differs significantly with previous state-of-the-arts in the sense that all three tasks are decoupled fairly and learned collaboratively using our proposed unified offset representation.

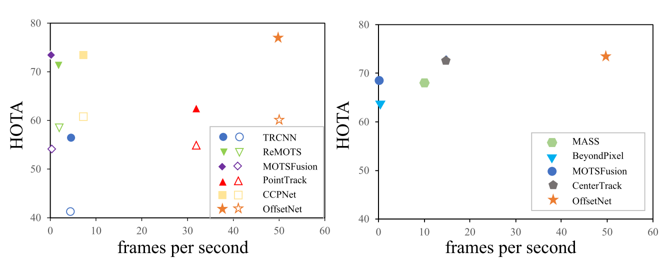

Fig. 3. Comparison between our proposed OffsetNet and the state-of-the-art MOTS(left) and MOT(right) methods. In the left subfigure, the filled symbols and the hollow symbols denote the results for cars and for pedestrians respectively. OffsetNet surpasses all prior works while running significantly faster at 50FPS.

better efficiency over utilizing high dimensional ReID features. Meanwhile, both bounding box and instance mask can be obtained via predicting intra-frame pixel offsets towards the object center in advanced bottom-up frameworks [8], [9]. We believe offset-based representation can naturally be extended to MOTS. However, since precise segmentation requires multi-scale intra-frame context representation while robust tracking relies heavily on inter-frame long-range correspondence, realizing both advantages in one network is not easy.

To tackle the above issues, we present a unified offset-based method for MOTS, named *OffsetNet*, which is able to produce tracking associations, amodal bounding boxes and instance masks simultaneously. Fig. 1 illustrates the concept of our proposed offset-based method. Note that compared with inmodal bounding box, amodal bounding box reflects the intrinsic size of the occluded instance and thus is important for tracking and feature learning. We verify through experiments that adding amodal bounding box prediction benefits both tracking and segmentation results. *OffsetNet* is an encoder-decoder based architecture, with three light-weight decoders accounting for instance localization and offset prediction, respectively. Moreover, in order to facilitate spatial-temporal feature aggregation in one compact encoder which is shared by the decoders, we further equip *OffsetNet* with a novel memory-enhanced linear self-attention (MELSA) block. Taking the advantage of MELSA, we further add a novel cross-frame offsets prediction mechanism that is able to improve occlusion handling. Finally, we apply a consistency loss to align features between decoders at multiple scales to enable mutual learning between decoders. Fig. 2 illustrates the difference between *OffsetNet* with previous state-of-the-art MOTS frameworks.

Without bells and whistles, as shown in Fig. 3, *OffsetNet* outperforms prior methods on both MOTS and MOT datasets with better efficiency. We hope our proposed method can server as a strong baseline for real-time MOTS and encourage practical applications.

The contributions of our paper can be summarized as:

- We present a simultaneous tracking, instance segmentation and detection framework, named *OffsetNet*. All three tasks are well-organized in a unified offset-based representation framework, making *OffsetNet* the first one-stage network that addresses the three challenging tasks.
- We equip *OffsetNet* with a compact yet highly effective encoder, in which non-local context and temporal dependencies are realized by a novel MELSA block.
- We further enhance offset-based tracking to improve occlusion handling via predicting tracking offsets across multiple frames.
- Extensive experimental comparisons show *OffsetNet* surpasses prior MOTS and MOT-based methods.

II. Related Work

Multi-Object Tracking: Recently, Multi-Object Tracking(MOT) [10], [11], [12], [13], [14] mainly follows a tracking-by-detection paradigm. MeMOT [15] maintains a temporal memory buffer for the embedding vectors of tracked objects and aggregates them by cross attention. STP [16] designs a model consisting of a society of classifiers to detect different tracker parts of the objects. They enhance the robustness of detection through the co-occurrence of a large number of smaller tracker parts. These methods rely heavily on the detection results and thus are not optimized in an end-to-end manner. More recently, some works [8], [17], [18], [19] pay attention to the joint training of detection and tracking. FairMOT [20] presents an efficient approach to learn detection and Reid features jointly in a unified network. However, existing methods are designed for realizing tracking and detection while do not support high-quality segmentation prediction in the same network.

On the other hand, occlusion handling is a long-standing problem in MOT. One common solution is to rely on ReID features to perform long-term data association [13], [21]. However, the ReID features are prone to interference due to occlusion. Some methods perform tracklet-level associations for associating objects across frames [22], [23], thus are hard to apply to real-time tasks. Gao et al. [24] categorize occlusion into inter-object occlusion and obstacle occlusion then handles

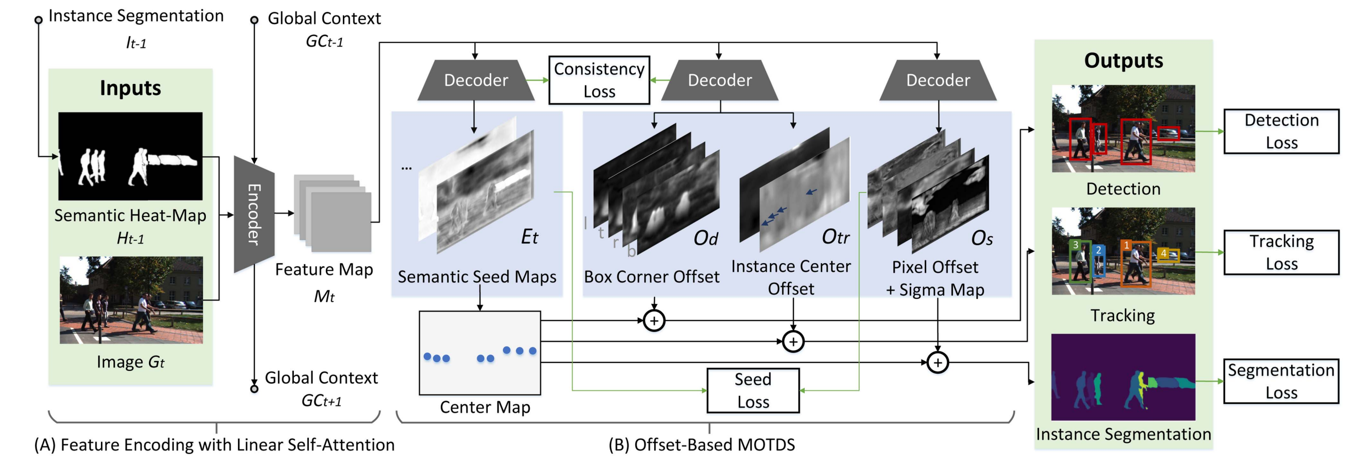

Fig. 4. The overall pipeline of our proposed OffsetNet for real-time multiple object tracking, amodal bounding box detection and instance segmentation. Inputs are the instance segmentation $\hat{I}_{t-1}$ and the current image I_t. Outputs are MOTS results, including amodal bounding boxes, tracking association, and instance mask. (A) Feature encoding module, which encodes inputs and the global context into the feature map M_t. (B) Offset-Based MOTS, which decodes M_t into three offsets, including the box corner offset O_d, the instance center offset O_{tr}, and the pixel offset O_s. Besides, instance localization is proposed to generate semantic seed maps E_t and the center map for producing the final MOTS results.

respectively. Stadler et al. [25] performs an occlusion handling strategy, which applies a linear motion model to those occluded tracks and thus is not robust for objects with nonlinear motions. Yang et al. [26] introduces a vectorial occlusion variable to solve the mutual occlusion, including an observation likelihood model and the occlusion prior using Markov Random field. Generally, existing occlusion handling research focuses on offline methods, which overlooks solving occlusions for real-time systems.

Multi-object tracking and segmentation: Multi-object tracking task is extended to multi-object tracking and segmentation(MOTS) in TrackRCNN [1] which also offers a baseline model with enhanced Mask-RCNN. Recently, some works [5], [6] further propose methods that utilize 3D perceptions as tracking clues. Besides, PointTrack [2] builds a novel online MOTS method that introduces a PointNet [27] like feature extraction strategy. MOTSNet [3] proposes a two-stage network and a training data generation pipeline for MOTS. The methods mentioned above [1], [2], [3] are basically two-stage frameworks, real-time and one-stage MOTS method is far from thoroughly researched. The latest VOS (video object segmentation) method STEM-Seg [4] models a video clip as a single 3D spatial-temporal volume, and proposes to segment instances across space and time in a single stage. However, this heavy framework operates with low frame rates (around 7 FPS). Recently, TransTrack [28] and TrackFormer [29] provide novel MOT/MOTS pipelines which leverage self-attention [30] and DETR [31]. PCAN [32] distills a space-time memory into a set of prototypes and then employs cross-attention to retrieve rich information from different frames. However, the quadratic time complexity of the self-attention module still limits their applications for real-time systems. More recently, CCP-Net [7] proposes the first one-stage and online MOTS which replaces the secondary ReID feature extraction sub-network by a simple spatial max-pooling operation. However, CCPNet achieves good segmentation results via CCP strategy but is weak in tracking.

Efficient Attention: There are three common categories in the practice of efficient attention mechanisms. The first category [31], [33], [34], [35] adopts a predefined sparse attention pattern on keys while the second category [36], [36], [37] further incorporates a data-dependent sparse attention. Besides, the third category [38], [39] explores the low-rank property in self-attention. Recent MOT studies [29], [40], [41] propose spatial and temporal attention mechanisms to associate a set of object bounding boxes and tracklets. These designs incur high time complexity owing to intricate spatial and temporal attention designs. Additionally, they do not effectively combine detection, tracking, and segmentation, resulting in a lack of collaboration between these distinct tasks. Our proposed MELSA which incorporates efficient linear self-attention [42] falls into the third category while further extending efficient spatial-temporal feature aggregations and task collaborations.

Offset-based Visual Recognition: Offset-based representation is widely used in various vision tasks, we provide a very brief review of related methods. For object detection, CenterNet [8] proposes an offset-based framework for both 2D and 3D detection. It also generalizes to pose estimation. For tracking, offset-based approaches can be regarded as sparse motion predictions in CenterTrack [8]. However, one major drawback of CenterTrack is that it only considers tracking offsets between adjacent frames thus seriously limiting performance when occlusion occurs. For segmentation, instance mask can be obtained via predicting intra-frame pixel offsets towards the object center in advanced bottom-up framework [9]. Hence, most existing offset-based research focuses on solving individual tasks.

III. OffsetNet

A. Overview

We propose a unified offset-based network, called *OffsetNet* for MOTS task. *OffsetNet* employs an encoder-decoder framework, with three light-weight decoders accounting for multi-tasking outputs. All decoders share the feature generated by the same encoder module. Fig. 4 illustrates our proposed overall pipeline.

At each time step t, our proposed *OffsetNet* takes the current image I_t and the semantic Heat-map H_{t-1} as input. The H_{t-1}

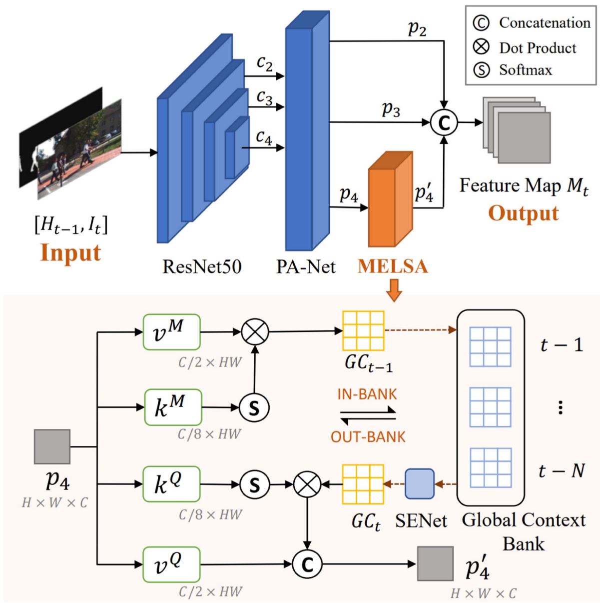

Fig. 5. *Above: Encoder.* A detailed pipeline of the encoder. c_2 and p_2 represent the low-level features, while c_4 and p_4 represent the semantic features. Note that we only apply MELSA for p_4, in order to encode semantic information into the global context bank, and keep the network efficiency. *Below: Memory Enhanced Linear Self-Attention (MELSA).* It enjoys two preferable properties: First, we leverage linear self-attention [42] (linear complexity $\mathcal{O}(H * W * C^2)$) to efficiently capture non-local contextual cues. Second, we employ the global context bank to supply non-local contextual cues for better occlusion handling. For the details, please see Section III-B.

is generated from instance segmentation results of the previous frame $\hat{I}_{t-1}$. These inputs are fed into the encoder to perform feature encoding. In the encoder, a memory bank is utilized to store temporal information to facilitate long-term tracking. The memory bank is designed as a FIFO queue to store the non-local contextual cues extracted from recent N frames. With the memory bank and a proposed self-attention mechanism, the encoder encodes inputs and the global context into the feature map M_t. The details of the feature encoding process are elaborated in Section III-B. With the feature maps M_t from the encoder, three decoders are designed to generate a center map C_t and three different offset maps i.e., pixel offset O_s, box corner offset O_d and instance center offset O_{tr} for segmentation, amodal object detection, and tracking tasks respectively. The details of these decoders and offset maps generation are further depicted in Section III-C. Finally, additional losses are introduced to achieve mutual collaborations of these decoders for multi-task training in Section III-D.

B. Feature Encoding

Towards real-time MOTS, building a compact encoder architecture is challenging and crucial to serve as the common backbone of the subsequent decoders. Also, in order to carry out object tracking, especially to tackle the occlusion issue in MOTS, it is important to comprehensively analyze the historical state of each object. In addition, the instance segmentation further requires exploration of spatial relations over the image. To design an encoder capable of aggregating information over both spatial and temporary domains, we marry self-attention mechanism with a global context bank to build a memory-enhanced feature encoding module which is illustrated in Fig. 5 in detail.

The input is the combination of I_t and the previous segmentation heatmap H_{t-1}. H_{t-1} is calculated by placing a Gaussian distribution at the center of each instance in $\hat{I}_{t-1}$ and background pixels are set to zero,

$$H_{t-1}(x,y) = \mathbb{1}(x,y)\sum_{i=1}^{N} exp\left(-\frac{\left(x-c_x^i\right)^2}{\left(\sigma_x^i\right)^2} - \frac{\left(y-c_y^i\right)^2}{\left(\sigma_y^i\right)^2}\right), \tag{1}$$

where x, y are the position of a pixel in H_{t-1}. c_x^i and c_y^i are the position of center P_i of the ith instances S_i in $\hat{I}_{t-1}$ and N denotes the number of instances. $\sigma_x^i = b_w^i/6 \times \alpha$ and $\sigma_y^i = b_h^i/6 \times \alpha$ are based on width b_w^i and height b_h^i of segmentation box of S_i and a fixed parameter $\alpha = 10$. $\mathbb{1}$ is the indicator function to classify the object and background on H_{t-1}.

We utilize ResNet50 together with PANets [43] to generate multi-scale features as p_2, p_3, p_4, respectively. Then, the semantic feature p_4 is fed into our proposed Memory Enhanced Linear Self-Attention(MELSA) block. Finally, the output p_4' of the MELSA block is combined with p_2 and p_3 to get the final feature map M_t.

The architecture of the MELSA block is illustrated in the lower part of Fig. 5. First, we leverage the self-attention module to capture spatial non-local context cues. Since the original self-attention [30] module has quadratic time complexity $\mathcal{O}((H * W)^2 * C)$, we adopt the recent accelerated architecture that reduces to linear complexity $\mathcal{O}(H * W * C^2)$ [42]. Here H, W denotes the height and width of the feature map while C denotes the channel number. Furthermore, to utilize sequential information from previous frames, we develop a light-weight memory bank for storing global context GC_{t-1} to GC_{t-N}, where N is the bank size. Then, we concatenate all the global context matrix in the memory bank and perform channel-wise attention (via a small SE-Net [44]) to compute GC_t for the current frame. The main structure of the small SE-Net is a single SE-layer and the details can be found in our supplementary material.

C. Decoders for Offset-Based MOTS

In order to better exploit the commonality of segmentation, detection and tracking, we design a unified MOTS architecture with three decoders. As shown in Fig. 4(B), the three tasks can be truly decoupled, which makes the proposed *OffsetNet* efficient. The design can also ensure collaboration between tasks since these decoders share the same encoder and the consistency loss is equipped between decoders. The pixel offset, the box corner offset, and the instance center offset are derived from these decoders. Specifically, the instance center offset takes advantage of the proposed MELSA block, hence it is expected to have the ability to resist occlusion. The overall network architecture is illustrated in Fig. 4(B). The input of the decoders is the feature map M_t which is introduced in Section III-B, and the outputs are the MOTS results. We further illustrate these modules as follows.

Instance Localization: The instance localization module is proposed for localizing the center of each instance, which is defined as a center map. Given the feature map M_t as input, a seed map decoder $F_{seed}(*)$ is designed to generate a semantic seed map E_t. The seed map E_t denotes the probabilities of pixels as the center of an instance. Afterward, the center map is able to be generated from the semantic seed map using the clustering algorithm, which is borrowed from [9]. The details are shown in Algorithm 1.

Pixel Offset: The pixel offset describes the displacements between the location of each pixel and their corresponding instance center. Specifically, M_t is fed into the segmentation decoder $F_s(*)$ to get the pixel offset O_s, where each value stands for the x or y displacements at each pixel. Besides, $F_s(*)$ also generates a sigma map σ to serve as the clustering margin for each object during the clustering process, which is illustrated in Algorithm 1.

Box Corner Offset: Similarly, the box corner offset describes the displacement of the $ltrb$ (left, top, right, bottom) of each bounding box with each corresponding instance center. We utilize the detection-tracking decoder $F_{dt}(*)$ to produce the box corner offset map O_d, which represents the $ltrb$ displacement at each pixel.

Instance Center Offset: Different from traditional MOT methods which perform instance tracking according to the offset between the center of the rigid bounding boxes, *OffsetNet* realizes the tracking offset according to the center of segmentation masks. Specifically, the instance center offsets O_{tr} share the decoder $F_{dt}(*)$ with the box corner offsets O_d. In O_{tr}, each value stands for the x or y displacement across frames.

Post processing: After we get the center map and the three offset maps, a post-process strategy is used in order to produce amodal bounding boxes, instance segments and tracking results. Specifically, amodal bounding boxes can be generated by the summation of the center map and the box corner offset. Instance segments can be calculated by the subtraction of the pixel offset and the offset value from each instance center, followed by thresholding to cluster the pixels from the same instance as shown in Algorithm 1. The tracking association is generated via the summation of the instance center offset and the center map, which is able to localize the corresponding location of the instance from previous frames. Then, a greedy algorithm is used to get the optimal tracking association for generating the tracking results as detailed in Algorithm 2.

Algorithm 1. Clustering Algorithm.

Input :
Semantic Seed Map E_t;
Pixel Offset Map O_s;
Sigma Map σ ;

Output:
Instance Segmentation S_t ;
Center Map C_t;

Initialize $S_t \leftarrow 0$;
Initialize $count \leftarrow 0$ // record number of instances;
Initialize $mask \leftarrow E_t > thres_{seed}$ // the unclustered area ;
repeat
  $seeds, indexes \leftarrow topk(E_t[mask])$ // find the local maximum of semantic seed map ;
  if $max(seeds) < thres_{seg}$ **then**
    break ;
  end
  $pcount \leftarrow sum(seeds \geq thres_{seg})$
  $c \leftarrow E_t[indexes]$ // get the center of proposal ;
  $\phi \leftarrow exp(-\frac{\|O_s - c\|^2}{2\sigma^2})$;
  $proposal \leftarrow \phi > thres_{prop}$ // get the instance proposal ;
  $S_t[proposal] \leftarrow count$ // assign an identity to proposal ;
  $count \leftarrow count + pcount$;
  $mask[proposal] \leftarrow 0$;
until *all pixels are clustered*;
$C_t \leftarrow$ instance center of S_t ;

D. Training

In order to remain simple and effective, the proposed *OffsetNet* is trained in an end-to-end paradigm. To be specific, five loss functions are designed for supervision, including the segmentation loss $\mathcal{L}_{\text{seg}}$, the detection loss $\mathcal{L}_{\text{det}}$, the tracking loss $\mathcal{L}_{\text{track}}$, the seed loss $\mathcal{L}_{\text{seed}}$, and the multi-scale consistency loss $\mathcal{L}_{\text{MC}}$.

Segmentation Loss: Denote the instance set at frame t as $\mathbb{S} = \{S_1, \ldots, S_N\}$, and the centroid of each instance as $P_k = \frac{1}{N}\sum_{p\in S_k} p$. Typically, O_s is learned using a segmentation loss $\mathcal{L}_{\text{seg}}$ with direct supervision:

$$\mathcal{L}_{\text{seg}} = \sum_{k=1}^{N}\sum_{p\in S_k} \max\left(||\,(p + O_s(p)) - \hat{P}_k|| - \delta, 0\right) \tag{2}$$

where p refers to one pixel location in S_k, $\hat{P}_k$ is the ground truth of the k-th instance centroid, and δ is the hyper-parameter.

Detection Loss: The loss $\mathcal{L}_{\text{det}}$ is designed by

$$\mathcal{L}_{det} = \frac{1}{N}\sum_{k=1}^{N} |O_d(k) - \hat{O}_d(k)| \tag{3}$$

where $O_d(k)$ denotes the box corner offset vector from the k-th instance center, and $\hat{O}_d(k)$ denotes the corresponding ground truth.

Tracking Loss: The loss $\mathcal{L}_{\text{track}}$ is designed as

$$\mathcal{L}_{track} = \frac{1}{N}\sum_{i=1}^{N} |O_{tr}(k) - \hat{O}_{tr}(k)| \tag{4}$$

where $O_{tr}(k)$ denotes the tracking offset vector for the k-th instance, and $\hat{O}_{tr}(k)$ denotes the corresponding ground truth, which is the euclidean distance between the centroid of each instance across frames.

Algorithm 2. Cross-Frame Instance Association.

Input :
Alive tracked objects $T_a = \{(frame, id, c)_i\}_{i=1}^{M}$,M is the length of alive tracked objects. ;
Instance Segmentation S_t;
Instance Center Offset Map O_{tr};
Output:
Tracked objects in the current frame T_t;

Initialize $T_t \leftarrow \emptyset$;
Initialize $P \leftarrow \emptyset$ // matched tracking pairs;
$N \leftarrow$ number of instances in S_t ;
$C_t \leftarrow$ instance center of S_t ;
$C_a \leftarrow \{for\ c_i\ in\ T_a\}_{i=1}^{M}$;
$D \leftarrow C_t + O_{tr}$ // get the predict center of last frame ;
$W \leftarrow D - C_a$ // compute the distance matrix ;
repeat
  $i, j \leftarrow argmin(W)$;
  if $W(i, j)$ *is available* **then**
    $P \leftarrow P \cup \{(i, j)\}$;
    set row i of W as unavailable;
    set column j of W as unavailable;
  end
until *No available value in* W;
for *each* (i, j) *in* P **do**
  $T_t \leftarrow T_t \cup (t, T_a[j].id, C_t[i])$ // update the alive track;
end
for $i \leftarrow 1$ *to* N *and not in* P **do**
  $T_t \leftarrow T_t \cup (t, nextID, C_t[i])$ // create a new track;
end

Seed Loss: In order to obtain a seed map that is able to reflect the distribution of instance centers, the seed loss $\mathcal{L}_{\text{seed}}$ is designed using the pixel offset as self-supervision. Specifically, it is defined by

$$\mathcal{L}_{\text{seed}} = \frac{1}{N}\sum_{k=1}^{N} \mathbb{1}_{\{p\in S_k\}} \|E_t(p) - \phi_k(p)\|^2 + \mathbb{1}_{\{p\in \text{bg}\}} \|E(p)\|^2 \tag{5}$$

where $E_t(p)$ is the location p in seed map E_t, $p \in bg$ means that pixel belongs to the background. $\phi_k(p)$ denotes the value of Gaussian distribution at location p based on O_s. Similar to (1), this Gaussian distribution is placed on the center P_k of S_k. The variances of Gaussian distribution are taken from the sigma map σ. Note that σ is also supervised as is shown in Algorithm 1.

Multi-Scale Consistency Loss: To enhance the correlation guidance between decoders, we additionally design a multi-scale consistency loss, which is applied to the decoder $F_{seed}(*)$ and decoder $F_{dt}(*)$. Denote the feature map of $F_{seed}(*)$ at the k-th scale as $\mathbf{F}_k^s \in \mathbb{R}^{h_k\times w_k\times d_k}$, and the feature map of $F_{dt}(*)$ at the k-th scale as $\mathbf{F}_k^d \in \mathbb{R}^{h_k\times w_k\times d_k}$. Therefore, the correlation R_k between $F_{seed}(*)$ and $F_{dt}(*)$ at the k-th scale is calculated by

$$R_k(i, j) = \frac{(f_i^s)^T f_j^d}{\|f_i^s\|\,\|f_j^d\|} \tag{6}$$

where f_i^s and f_j^d are the row vectors of $\mathbf{F}_k^s$ and $\mathbf{F}_k^d$ respectively. Then, feature embedding is conducted for $\mathbf{F}_k^s$ and $\mathbf{F}_k^d$, resulting in $Q_k^s = \mathbf{F}_k^s\mathbf{W}_{s,k}$ and $Q_k^d = \mathbf{F}_k^d\mathbf{W}_{d,k}$, where $\mathbf{W}_{s,k}, \mathbf{W}_{d,k} \in \mathbb{R}^{d_k\times 1}$ are linear learnable parameters. Thus, the multi-scale consistency loss is designed as

$$\mathcal{L}_{\text{MC}} = -\sum_{k=1}^{K}\sum_{i=1}^{h_k\times w_k}\sum_{j=1}^{h_k\times w_k} \log\left[Q_{s,k}(i)R_k(i,j)Q_{d,k}(j)\right]. \tag{7}$$

Here, K is the number of multi-scales, and we choose $K = 3$ in our experiment.

Finally, the loss function $\mathcal{L}$ is defined by the weighted summation of all loss terms,

$$\mathcal{L} = \mathcal{L}_{\text{seg}} + \lambda_{det}\mathcal{L}_{\text{det}} + \lambda_{track}\mathcal{L}_{\text{track}} + \lambda_{seed}\mathcal{L}_{\text{seed}} + \mathcal{L}_{\text{MC}}, \tag{8}$$

where λ_{det}, λ_{track} and λ_{seed} represent the weights for $\mathcal{L}_{\text{det}}$, $\mathcal{L}_{\text{track}}$ and $\mathcal{L}_{\text{seed}}$ respectively.

IV. Experiments

Our method is evaluated on KITTI MOT dataset [45], KITTI MOTS [1] dataset, APOLLO MOTS [2] dataset and MOTSChallenge [46] benchmark. In addition, the ablation study is also conducted to demonstrate the effectiveness of our design.

A. Datasets and Evaluation Metrics

KITTI tracking dataset includes KITTI MOT [45] and KITTI MOTS [1]. To be specific, KITTI MOT has complete 2D bounding boxes annotation for cars and pedestrians, and consists of 21 training sequences and 29 test sequences. KITTI MOTS provides pixel-level mask annotations based on KITTI tracking images to evaluate MOTS tasks.

MOTSChallenge provides mask annotations for only pedestrians in 4 train and test sequences. Since the segmentation masks should be clearly visible, only large objects are annotated.

APOLLO MOTS provides mask annotations for cars in 85 train sequences and 84 test sequences. It is built on the ApolloScape dataset [47] in which pedestrians are much fewer than cars. APOLLO MOTS has 2.5 times more crowded cars than KITTI MOTS.

Evaluation metrics For a fair comparison with previous methods, all datasets mentioned above are evaluated with standard CLEAR MOT [48] evaluation metrics or MOTS-extended metrics [1]. Recently, High Order Metric (HOTA) is proposed for striking a balance between detection/segmentation and association in tracking-related tasks [49]. We also adopt this new metric in the evaluations on the KITTI test benchmark. For MOT, we report Multiple Object Tracking accuracy (MOTA), Multiple Object Tracking Precision (MOTP) and Identity Switches (IDS). For MOTS, we report soft Multiple Object Tracking and Segmentation Accuracy (sMOTSA), and Multiple Object Tracking and Segmentation Accuracy (MOTSA). For time consumption

TABLE I
EVALUATION ON THE KITTI MOTS VALIDATION SET

	Pretrained	Cars				Pedestrians				Time ↓
		sMOTSA ↑	MOTSA ↑	MOTA ↑	IDS ↓	sMOTSA ↑	MOTSA ↑	MOTA ↑	IDS ↓	
MRCNN+maskprop	KINS	75.1	86.6	-	-	45.0	63.5	-	-	-
TRCNN [1]	KINS	76.2	87.8	-	93	46.8	65.1	-	78	0.5s
STEM-Seg [4]	MS-COCO [51]	72.7	83.8	-	76	50.4	66.1	-	14	-
MOTSNet [3]	MV [52]	78.1	87.2	-	-	54.6	69.3	-	-	-
PCAN [32]	-	-	89.6	-	-	-	66.4	-	-	-
BeyondPixel [53]	-	84.9	93.8	-	97	-	-	-	-	3.96s
MOTSFusion [6]	-	85.5	94.6	-	35	-	-	-	-	4.04s
PointTrack (w/o TC)	KINS	82.9	92.7	-	25	61.4	76.8	-	21	28ms
PointTrack [2]	KINS	85.5	94.9	-	22	62.4	77.3	-	19	28ms
CenterTrack [8]	Crowdhuman [54]	-	-	88.9	31	-	-	65.7	25	68ms
PointTrackV2 [55]	KINS	**86.2**	**95.5**	-	18	63.7	78.5	-	22	22ms
OPITrack [56]	-	85.5	94.9	-	22	62.4	77.3	-	19	45ms
Ours	KINS	85.9	95.0	**92.9**	**16**	**63.8**	**80.1**	**68.3**	**13**	**19ms**

We clearly show our advantage of the unified model over other multi-stage methods, since it only costs about 19ms for inference per frame while both other MOT and MOTS methods cost much more time for inference per frame. 'Pretrained' denotes the method is pre-trained on other datasets other than ImageNet [57].

TABLE II
EVALUATION ON THE KITTI MOTS TEST SET

	Pretrained	Cars						Pedestrains					
		HOTA ↑	DetA ↑	AssA ↑	sMOTSA	MOTSA	IDS ↓	HOTA ↑	DetA ↑	AssA ↑	sMOTSA	MOTSA	IDS ↓
TRCNN	KINS	56.63	69.90	46.53	67.00	79.60	692	41.93	53.75	33.84	47.30	66.10	481
MOTSFusion	-	73.63	84.10	73.63	75.00	84.10	201	54.04	60.83	49.45	58.70	72.90	279
PointTrack	KINS	61.95	90.90	61.95	78.50	90.90	346	54.44	62.29	48.08	61.50	76.50	176
ReMOTS [58]	-	71.61	78.32	65.98	75.92	-	716	58.81	67.96	52.38	65.97	-	391
CCPNet [7]	KINS	73.61	**84.47**	64.58	**84.47**	**94.40**	**197**	**60.50**	**71.35**	52.50	**70.16**	**85.85**	275
OPITrack	-	73.04	79.44	67.97	78.02	-	542	60.38	62.45	**60.05**	61.05	-	234
PointTrackV2	KINS	67.28	-	-	82.20	92.20	298	56.67	-	-	67.20	83.00	184
Ours	KINS	**76.83**	79.42	**74.81**	80.88	91.33	270	59.93	65.04	55.79	63.44	79.99	**153**

Our OffsetNet surpasses other methods by a large margin. Especially in the HOTA metric, we have outperformed PointTrack by 14.3%.

analysis, all of the experiments are conducted in NVIDIA Tesla P40 24GB with Pytorch [50].

B. Implementation Details

Most previous works pre-train their networks on the KINS dataset [59] [1], [2], [60], [61], we hence also employ this strategy. Since the KINS dataset provides only static images, pseudo-moving video clips are generated for the training purpose. To this end, we crop the image of KINS to 224×800 according to the center of instances, and generate a three-frame synthetic video clip with random affine transforms for each crop to mimic the object movement in the wild. To increase the robustness to severe-occluded cases, we first construct an instance library containing all labeled moving objects in the training set. During the training process, randomly selected instances from this library are pasted onto the training clips to create occlusion data augmentation.

We use a variant of ResNet-50 [62] as our backbone. The channel number of the encoder output M_t is set to 256, while the channel number of the multi-scale consistency module are set to {256, 128, 64}. The network is trained with RAdam optimizer [63] using weight decay as $1e^{-4}$. For KITTI and MOTSChallenge, we fine-tune our network with multi-task loss for 20 epochs at a learning rate of $5e^{-6}$ with exponential learning rate decay. For APOLLO MOTS, we train from scratch for 50 epochs at a learning rate of $2.5e^{-4}$ with multi-step learning rate decay. Besides, the value of λ_{det} and λ_{track} are set to 0.1 throughout the training phase while the value of λ_{seed} is 20 for pretraining and increases to 200 during finetuning.

C. Comparison With State-of-the-Arts

KITTI MOTS As shown in Table I, we compare our proposed *OffsetNet* with other state-of-the-art MOTS methods on the validation set of KITTI MOTS. *OffsetNet* outperforms all state-of-the-art methods in terms of both MOTS quality and inference time efficiency. Specifically, *OffsetNet* outperforms PointTrack by 0.4% and 1.4% in terms of sMOTSA for cars and pedestrians, respectively. Notably, PointTrack adopts temporal consistency loss where the optical flow is introduced as prior knowledge. In contrast, our model only exploits conventional MOT annotations without optical flow.

For time consumption analysis, we calculate the sum of tracking, detection and segmentation time for comparison. The experimental results clearly demonstrate the advantage of using our unified model against multi-stage competitors. Specifically, in comparison with PointTrack [2], our *OffsetNet* is more efficient (32% faster) since it enjoys the efficiency of single stage framework without relying on the two stage instance-wise embedding operation. In general, *OffsetNet* only costs 19ms for the whole inference process per frame for all three tasks, which is not only more efficient than previous MOTS methods but also super-passes previous MOT methods like CenterTrack.

TABLE III
EVALUATION ON THE KITTI MOT TEST SET

	MOTA ↑	MOTP ↑	HOTA ↑
AB3D [64]	83.84	85.24	69.81
BeyondPixel	84.24	**85.73**	63.75
mmMOT [65]	84.77	85.21	62.05
MOTSFusion	84.83	85.21	68.74
MASS [66]	85.04	85.53	68.25
CenterTrack	89.44	85.05	73.02
DEFT [67]	88.38	84.55	74.23
TripletTrack [68]	84.32	-	73.58
Ours	**90.31**	85.15	**74.83**

It is obvious that our OffsetNet is completely ahead of other MOT methods.

TABLE IV
EVALUATION ON MOTSCHALLENGE

	sMOTSA ↑	MOTSA ↑
MOTDT [69] + MG	47.8	61.1
MHT-DAM [70] + MG	48.0	62.7
jCC [71] + MG	48.3	63.0
FWT [72] + MG	49.3	64.0
TrackRCNN	52.7	66.9
MOTSNet	56.8	69.4
PointTrack	58.1	70.6
TrackFormer	58.7	-
OPITrack	63.5	76.5
PointTrackV2	62.3	76.8
Ours	**59.2**	**71.1**

"+MG" denotes the mask generation with a domain fine-tuned Mask-RCNN.

To further demonstrate the robustness and effectiveness of *OffsetNet*, we also report the result of KITTI MOTS test benchmark. As shown in Table II, *OffsetNet* outperforms PointTrack by 14.3% in terms of HOTA. In the comparison with the state-of-the-art MOTS method CCPNet [7], *OffsetNet* is better on cars (3.2%) while slightly falls behind on pedestrians (0.6%) in terms of HOTA. Though the overall performance is competitive, we discover that *OffsetNet* and CCPNet pay different attends on tracking and segmentation. More specifically, *OffsetNet* is better at tracking as the AssA score is significantly higher than CCPNet. In contrast, CCPNet obtains better segmentation results with their proposed continuous copy-paste strategy thus the DetA scores are better.

KITTI MOT Since many MOT method performs tracking based on the bbox detections without doing segmentation. In order to conduct more comprehensive comparisons for detection and tracking quality, we compare *OffsetNet* with leading MOT methods on the KITTI MOT dataset. As shown in Table I, our method surpasses all the other leading MOT methods which indicates our proposed unified multi-tasking network is able to obtain high-quality bbox detection and tracking results.

MOTSChallenge Compared with KITTI dataset, scenes in MOTSChallenge are more crowd, and hence more challenging. We follow the same training strategy (leaving-one-out) as previous works [1], [3] and results are shown in Table IV. Our *OffsetNet* achieves state-of-the-art MOTS results in terms of sMOTSA and MOTSA, which indicates the great generalization of our proposed method.

TABLE V
ABLATION STUDY ON APOLLO MOTS VALIDATION SET

Tracker	Backbone	sMOTSA ↑	MOTSA ↑
DeepSort [73]	MRCNN	45.71	57.06
TRCNN	MRCNN	49.84	61.19
PointTrack	ERFNet [74]	70.76	80.05
PointTrackV2	RandLA [75]	72.24	81.54
Ours	ERFNet	71.32	80.95
Ours	Ours	**72.68**	**81.63**

With the same ERFNet as the backbone, the sMOTSA and MOTSA of our method surpass that of PointTrack by a clear margin. Meanwhile, the performance further increases by replacing ERFNet with ResNet in our proposed OffsetNet.

TABLE VI
COMPARISONS OF IDS ON KITTI AND APOLLO MOTS WITH DIFFERENT SEGMENTATION MASKS

Dataset	Seg.	Method	IDS(car) ↓	IDS(ped.) ↓
KITTI MOTS val	TRCNN	TRCNN	93	78
		PointTrack	46	30
		Ours	**31**	**25**
	Ours	PointTrack	25	22
		Ours	**16**	**13**
APOLLO MOTS	TRCNN	DeepSort	1263	-
		PointTrack	241	-
		Ours	**186**	-
	Ours	PointTrack	231	-
		Ours	**192**	-

OffsetNet shows obvious advantage in reducing ID switches. "Seg" means the method from which we obtained the segmentation mask.

TABLE VII
ABLATION STUDY OF MEMORY ENHANCED LINEAR SELF-ATTENTION (MELSA) ON KITTI MOTS VALIDATION SET

	sMOTSA ↑	MOTA ↑	IDS ↓
OffsetNet w/o MELSA	84.8	92.1	28
OffsetNet w/ MELSA	**85.9**	**92.9**	**19**

The comparison illustrates the effectiveness of the proposed MELSA.

D. Ablation Study

Compared to KITTI, the average car density of APOLLO MOTS is 5.65 and thus is more crowded [2]. Therefore, we adopt this challenging dataset in the ablation study to better investigate the effectiveness of our design.

Comparison of our methods with different backbones and trackers: We replace the backbone and trackers to validate the advantage of our design. As shown in Table V, both PointTrack and our *OffsetNet* are much more well-performed than the Mask-RCNN-based network on APOLLO MOTS. The performance of *OffsetNet* is also superior to the PointTrackV2, which adopts a newer backbone RandLA. To examine the effectiveness of our *OffsetNet*, we replace the original resnet-style backbone with ERFNet which is the same backbone as used in PointTrack, a better result (sMOTSA 71.32 and MOTSA 80.95) against PointTrack (sMOTSA 70.76 and MOTSA 80.05) is obtained. Meanwhile, the performance further increases by replacing ERFNet with ResNet. This demonstrates that our proposed *OffsetNet* achieves better results using the same backbone as

Fig. 6. *Top:* Partially-occluded instance in consecutive frames. *Bottom:* Fully-occluded instance in selected non-consecutive frames. Both in *Top* and *Bottom*, the red point indicates the center of the instance while the red arrow indicates the predicted tracking offset. *OffsetNet* with MELSA can predict more precise instance center offset in both conditions.

TABLE VIII
ABLATION STUDY ABOUT SOME CORE COMPONENTS ON THE KITTI MOTS VALIDATION SET TO VALIDATE THE EFFECTIVENESS OF MUTUAL COLLABORATIONS

ABP	MCG	MSCL	sMOTSA ↑	MOTA ↑	IDS ↓
	✓	✓	85.2	85.3	**19**
✓		✓	85.1	91.9	27
✓	✓		85.6	92.7	22
✓	✓	✓	**85.9**	**92.9**	**19**

ABP denotes the amodal bboxes prediction. Without ✓ for ABP here means we perform segmentation and tracking without amodal bboxes detection. MCG denotes using mask center-guided tracking (with ✓) rather than box center-guided tracking (without ✓). MSCL denotes the multi-scale consistency loss.

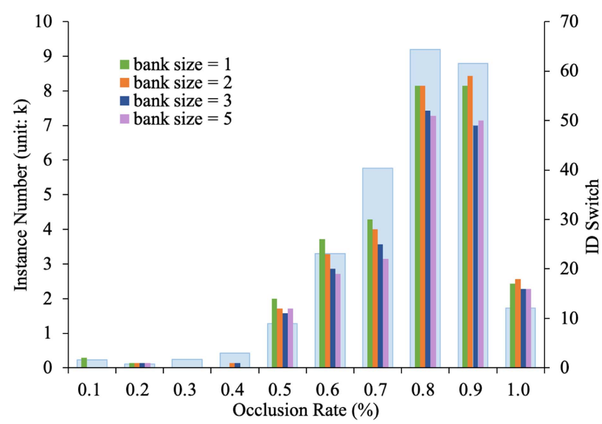

Fig. 7. We illustrate the number of id switch with respect to the ratio of occlusion evaluated on the APOLLO MOTS dataset corresponding to the right axis. The transparent blue bar indicates the number of instances in videos at each occlusion ratio corresponding to the left axis, and illustrates the proportionate impact on the overall performance of the trackers. The results clearly demonstrate that the memory mechanism is effective in severe-occluded conditions.

PointTrack and can also be compatible with better backbone architecture.

Comparison of our methods with different segmentation masks: In addition, we further evaluate the tracking performance of our method using different segmentation masks using the metric of ID Switches (IDS) on both KITTI MOTS validation set and APOLLO MOTS. As demonstrated in Table VI, based on the same segmentation results from TRCNN, the tracking offset produced by *OffsetNet* is able to produce the lowest IDS compared with the other methods on both KITTI-MOTS and APOLLO-MOTS dataset. Based on our generated segmentation masks, our method also defeats PointTrack by a clear margin in terms of IDS on the both datasets, which indicates a clear advantage of our method in reducing ID switches.

E. Effectiveness of MELSA

In Fig 6, we show how MELSA improves the tracking offset prediction significantly in both partially occluded and fully occluded conditions. Table VII shows a comparison between using MELSA module and using the multi-scale feature without MELSA in our proposed *OffsetNet*. When removing MELSA, the results under sMOTSA and MOTSA drop by 1.1% and 0.8 %, respectively, and the IDS of *OffsetNet* increase from 19 to 28. It demonstrates the effectiveness of MELSA to aggregate information over both spatial and temporary domains experimentally.

To demonstrate the performance improvements under severe occlusion conditions quantitatively, and to show the robustness of our method, we examine the performance of *OffsetNet* in terms of IDs using different bank sizes in MELSA. As shown in Fig 7, we evaluate the performance of instances with different occlusion rates in the APOLLO MOTS dataset. Our method shows superior performance in severe occlusion conditions. From this experiment, we see that a bigger bank size is helpful in handling occlusion and a reasonable balance between efficiency and performance is achieved when the bank size is set to 3.

Fig. 8. Left of the figure is bounding box guided and right of the figure is segmentation guided. The center point of the left is colored red which is not on the object itself while the center point of the right is colored purple. Instance center offsets are colored magenta.

F. Analysis of Mutual Collaborations Among Detection, Segmentation, and Tracking

We elaborate on mutual collaborations of different tasks from three perspectives: First, Compared to the traditional bounding box center-guided tracking in MOT methods, segmentation mask center-guided tracking is more reliable and is beneficial for generating reliable tracking offset predictions. Second, we conduct experiments to demonstrate that decoupling the all three tasks and simultaneous training with detection and tracking helps to enhance the quality of segmentation. Finally, we also report the performance without multi-scale consistency losses to show the collaborations of different decoders. The main results are shown in Table VIII.

Effectiveness of mask center-guided tracking Fig. 8 visualizes the major differences between bounding box center-guided tracking [8] and our proposed segmentation mask center-guided tracking in *OffsetNet*. As shown in Fig. 8, since the bounding box center usually gets occluded, adopting the center of the visible mask instead to serve as the starting point of the association offset vector is more reliable.

Further, we provide the comparisons between mask center-guided and bounding box center-guided tracking strategies in Table VIII. By replacing box-center-guided tracking with mask center-guided tracking, our proposed segmentation mask center-guided tracking improves 0.8% on sMOTSA, 1.0% on MOTA, and reduces 8 on IDS, respectively. This demonstrates that segmentation masks can be exploited to assist high-quality tracking.

Effectiveness of adding amodal bounding box detection We provide with the results of *OffsetNet* without amodal bbox prediction in Table VIII. As shown in the performance with and without ABP, the full version of *OffsetNet* with amodal detection surpasses segmentation-only version on all metrics by a clear margin. This comparison further reveals the effectiveness of adding amodal bbox detection and our proposed unified offset-based representation and the mutual learning strategy.

Effectiveness of multi-scale consistency loss As shown in Table VIII, we compare the performance of removing the multi-scale consistency loss from *OffsetNet*. The drop on sMOTSA and MOTA demonstrates the mutual collaborations from different decoders with multi-scale consistency loss.

V. Discussion and Conclusion

In this paper, we propose a unified real-time framework *OffsetNet* for multiple object tracking, detection and instance segmentation. Specially, a novel MELSA block is designed to aggregate the non-local context and temporal knowledge using a self-attention mechanism and global context memory bank. Furthermore, we predict tracking offsets to improve occlusion handling for multi-object tracking. Being the first compact solution, the proposed *OffsetNet* is capable of performing MOTS at 50 FPS while achieving significantly better performance compared with prior frameworks both in MOT and MOTS studies. Our method clearly sets up a new state-of-the-art.

Limitations: Our method still requires ground-truth labels for all tasks. This encourages us to explore training data-efficient MOTS framework in our future research.

Broader impacts: This paper studies a data-driven method for multi-object tracking, detection and segmentation. The labeling quality and sample distribution of the training data set will directly affect the performance of the algorithm and its generalization in actual scenarios, as such will reflect biases in testing scenarios including ones with negative societal impacts.

References

[1] P. Voigtlaender et al., "MOTS: Multi-object tracking and segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7942–7951.

[2] Z. Xu et al., "Segment as points for efficient online multi-object tracking and segmentation," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2020, pp. 264–281.

[3] L. Porzi, M. Hofinger, I. Ruiz, J. Serrat, S. R. Bulo, and P. Kontschieder, "Learning multi-object tracking and segmentation from automatic annotations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6846–6855.

[4] A. Athar, S. Mahadevan, A. Osep, L. Leal-Taixé, and B. Leibe, "STEm-Seg: Spatio-temporal embeddings for instance segmentation in videos," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2020, pp. 158–177.

[5] S. Qiao, Y. Zhu, H. Adam, A. Yuille, and L.-C. Chen, "ViP-DeepLab: Learning visual perception with depth-aware video panoptic segmentation," 2020, *arXiv: 2012.05258*.

[6] J. Luiten, T. Fischer, and B. Leibe, "Track to reconstruct and reconstruct to track," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 1803–1810, Apr. 2020.

[7] Z. Xu, A. Meng, Z. Shi, W. Yang, Z. Chen, and L. Huang, "Continuous copy-paste for one-stage multi-object tracking and segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 15323–15332.

[8] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2020, pp. 474–490.

[9] D. Neven, B. D. Brabandere, M. Proesmans, and L. V. Gool, "Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8837–8845.

[10] L. Leal-Taixé, C. Canton-Ferrer, and K. Schindler, "Learning by tracking: Siamese CNN for robust target association," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2016, pp. 33–40.

[11] S. Schulter, P. Vernaza, W. Choi, and M. Chandraker, "Deep network flow for multi-object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6951–6960.

[12] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *Proc. IEEE Int. Conf. Image Process.*, 2016, pp. 3464–3468.

[13] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, "Tracking without bells and whistles," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 941–951.

[14] Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang, "Towards real-time multi-object tracking," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2020, pp. 107–122.

[15] J. Cai et al., "MeMot: Multi-object tracking with memory," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 8090–8100.

[16] E. Burceanu and M. Leordeanu, "Learning a robust society of tracking parts using co-occurrence constraints," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2018, pp. 162–178.

[17] J. Peng et al., "Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2020, pp. 145–161.

[18] B. Pang, Y. Li, Y. Zhang, M. Li, and C. Lu, "TubeTK: Adopting tubes to track multi-object in a one-step training model," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6308–6318.
[19] Y. Wang, X. Weng, and K. Kitani, "Joint detection and multi-object tracking with graph neural networks," 2020, *arXiv: 2006.13164*.
[20] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "FairMOT: On the fairness of detection and re-identification in multiple object tracking," *Int. J. Comput. Vis.*, vol. 129, no. 11, pp. 3069–3087, 2021.
[21] S. Tang, M. Andriluka, B. Andres, and B. Schiele, "Multiple people tracking by lifted multicut and person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3539–3548.
[22] B. Wang, G. Wang, K. L. Chan, and L. Wang, "Tracklet association by online target-specific metric learning and coherent dynamics estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 3, pp. 589–602, Mar. 2017.
[23] Y. Zhang et al., "Long-term tracking with deep tracklet association," *IEEE Trans. Image Process.*, vol. 29, pp. 6694–6706, 2020.
[24] X. Gao and T. Jiang, "OSMO: Online specific models for occlusion in multiple object tracking under surveillance scene," in *Proc. 26th ACM Int. Conf. Multimedia*, New York, NY, USA, 2018, pp. 201–210. [Online]. Available: http://doi.acm.org/10.1145/3240508.3240548
[25] D. Stadler and J. Beyerer, "Improving multiple pedestrian tracking by track management and occlusion handling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 10958–10967.
[26] M. Yang, Y. Liu, L. Wen, Z. You, and S. Z. Li, "A probabilistic framework for multitarget tracking with mutual occlusions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1298–1305.
[27] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 652–660.
[28] P. Sun et al., "TransTrack: Multiple-object tracking with transformer," 2020, *arXiv: 2012.15460*.
[29] T. Meinhardt, A. Kirillov, L. Leal-Taixe, and C. Feichtenhofer, "TrackFormer: Multi-object tracking with transformers," 2021, *arXiv:2101.02702*.
[30] A. Vaswani et al., "Attention is all you need," 2017, *arXiv: 1706.03762*.
[31] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2020, pp. 213–229.
[32] L. Ke, X. Li, M. Danelljan, Y.-W. Tai, C.-K. Tang, and F. Yu, "Prototypical cross-attention networks for multiple object tracking and segmentation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 1192–1203.
[33] P. J. Liu et al., "Generating Wikipedia by summarizing long sequences," 2018, *arXiv: 1801.10198*.
[34] M. Zaheer et al., "Big bird: Transformers for longer sequences," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 17283–17297.
[35] A. Hassani, S. Walton, J. Li, S. Li, and H. Shi, "Neighborhood attention transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 6185–6194.
[36] N. Kitaev, Ł. Kaiser, and A. Levskaya, "Reformer: The efficient transformer," 2020, *arXiv: 2001.04451*.
[37] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," 2020, *arXiv: 2010.04159*.
[38] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma, "Linformer: Self-attention with linear complexity," 2020, *arXiv: 2006.04768*.
[39] K. Choromanski et al., "Rethinking attention with performers," 2020, *arXiv: 2009.14794*.
[40] Q. Chu, W. Ouyang, H. Li, X. Wang, B. Liu, and N. Yu, "Online multi-object tracking using CNN-based single object tracker with spatial-temporal attention mechanism," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4836–4845.
[41] J. Zhu, H. Yang, N. Liu, M. Kim, W. Zhang, and M.-H. Yang, "Online multi-object tracking with dual matching attention networks," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 366–382.
[42] Z. Shen, M. Zhang, H. Zhao, S. Yi, and H. Li, "Efficient attention: Attention with linear complexities," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2021, pp. 3531–3539.
[43] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8759–8768.
[44] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2011–2023, Aug. 2020.
[45] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.
[46] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," 2016, *arXiv:1603.00831*.
[47] X. Huang et al., "The apolloscape dataset for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 954–960.
[48] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The clear mot metrics," *EURASIP J. Image Video Process.*, vol. 2008, pp. 1–10, 2008.
[49] J. Luiten et al., "HOTA: A higher order metric for evaluating multi-object tracking," *Int. J. Comput. Vis.*, vol. 129, pp. 548–578, 2020.
[50] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8026–8037.
[51] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. 13th Eur. Conf. Comput. Vis.*, Zurich, Switzerland, Springer, 2014, pp. 740–755.
[52] G. Neuhold, T. Ollmann, S. Rota Bulo, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4990–4999.
[53] S. Sharma, J. A. Ansari, J. K. Murthy, and K. M. Krishna, "Beyond pixels: Leveraging geometry and shape cues for online multi-object tracking," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 3508–3515.
[54] S. Shao et al., "CrowdHuman: A benchmark for detecting human in a crowd," 2018, *arXiv: 1805.00123*.
[55] Z. Xu, W. Yang, W. Zhang, X. Tan, H. Huang, and L. Huang, "Segment as points for efficient and effective online multi-object tracking and segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 6424–6437, Oct. 2022.
[56] Y. Gao, H. Xu, Y. Zheng, J. Li, and X. Gao, "An object point set inductive tracker for multi-object tracking and segmentation," *IEEE Trans. Image Process.*, vol. 31, pp. 6083–6096, 2022.
[57] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.
[58] F. Yang et al., "ReMOTS: Self-supervised refining multi-object tracking and segmentation," 2020, *arXiv: 2007.03200*.
[59] L. Qi, L. Jiang, S. Liu, X. Shen, and J. Jia, "Amodal instance segmentation with kins dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3014–3023.
[60] S. W. Oh, J.-Y. Lee, K. Sunkavalli, and S. J. Kim, "Fast video object segmentation by reference-guided mask propagation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7376–7385.
[61] F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung, "Learning video object segmentation from static images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2663–2672.
[62] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[63] L. Liu et al., "On the variance of the adaptive learning rate and beyond," 2019, *arXiv: 1908.03265*.
[64] X. Weng, J. Wang, D. Held, and K. Kitani, "3D multi-object tracking: A baseline and new evaluation metrics," 2020, *arXiv: 1907.03961*.
[65] W. Zhang, H. Zhou, S. Sun, Z. Wang, J. Shi, and C. C. Loy, "Robust multi-modality multi-object tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 2365–2374.
[66] H. Karunasekera, H. Wang, and H. Zhang, "Multiple object tracking with attention to appearance, structure, motion and size," *IEEE Access*, vol. 7, pp. 104423–104434, 2019.
[67] M. Chaabane, P. Zhang, J. R. Beveridge, and S. O'Hara, "DEFT: Detection embeddings for tracking," 2021, *arXiv:2102.02267*.
[68] N. Marinello, M. Proesmans, and L. Van Gool, "TripletTrack: 3D object tracking using triplet embeddings and LSTM," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 4500–4510.
[69] L. Chen, H. Ai, Z. Zhuang, and C. Shang, "Real-time multiple people tracking with deeply learned candidate selection and person re-identification," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2018, pp. 1–6.
[70] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg, "Multiple hypothesis tracking revisited," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4696–4704.
[71] M. Keuper, S. Tang, B. Andres, T. Brox, and B. Schiele, "Motion segmentation & multiple object tracking by correlation co-clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 1, pp. 140–153, Jan. 2020.

[72] R. Henschel, L. Leal-Taixé, D. Cremers, and B. Rosenhahn, "Fusion of head and full-body detectors for multi-object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 1428–1437.

[73] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proc. IEEE Int. Conf. Image Process.*, 2017, pp. 3645–3649.

[74] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, "ERFNet: Efficient residual factorized convnet for real-time semantic segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 263–272, Jan. 2018.

[75] Q. Hu et al., "RandLA-Net: Efficient semantic segmentation of large-scale point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11108–11117.

Wei Zhang received the PhD degree from the University of Hong Kong, in 2017. He is currently a senior engineer with the Department of Computer Vision Technology at Baidu Inc. His current research interests include 2D/3D object detection, segmentation, and tracking. He is dedicated to the development of computer vision models for autonomous driving and intelligent transportation systems.

Jiaming Li received the BS degree from Sun Yat-sen University, Guangzhou, China, in 2020. He is currently working toward the MS degree with Sun Yat-sen University. His research interests include deep learning and computer vision, including semantic segmentation, and object detection.

Meng Xia received the master's degree from Sun Yat-sen University, Guangzhou, China, in 2022. His research interests include deep learning and computer vision, including instance segmentation and multi-object tracking.

Xu Gao received the MSc degree in technology of computer application from Peking University, in 2019. He is currently a senior engineer with the Autonomous Driving and Vehicle-Infrastructure Cooperation group, Baidu. His main research interests focus on multi-object tracking, trajectory prediction, and 3D object detection.

Xiao Tan received the PhD degree in computer vision from the University of New South Wales, Sydney, in 2014. His research interests include computer vision, pattern recognition, and image processing. He is currently with the Department of Comupter Vision, Baidu as a senior engineer. He served as a reviewer for ICCV, CVPR, ECCV, AAAI, IJCV, and etc.

Yifeng Shi received the master's degree in electrical engineering from Tianjin University, in 2017, with an emphasis on image processing. Currently, he is a senior engineer with Baidu and primarily responsible for the development of computer vision algorithms in the areas of autonomous driving and Vehicle-Infrastructure Cooperation. His main research interests include 2D/3D object detection and foundation model.

Zhenhua Huang received the PhD degree in the computer science from Fudan University, in 2008. He is currently a professor with the School of Computer Science at South China Normal University. His research interests mainly include deep learning, computer vision, recommender system, knowledge discovery and Big Data. Since 2004, he has published three books and more than 110 papers in various journals and conference proceedings.

Guanbin Li (Member, IEEE) received the PhD degree from the University of Hong Kong, in 2016. He is currently a full professor with the School of Data and Computer Science, Sun Yat-sen University. His current research interests include computer vision, image processing, and deep learning. He is a recipient of ICCV 2019 Best Paper Nomination Award. He has authorized and co-authorized on more than 200 papers in top-tier academic journals and conferences.