# Towards Grand Unification of Object Tracking

Bin Yan[1*], Yi Jiang[2,†], Peize Sun[3], Dong Wang[1,†],
Zehuan Yuan[2], Ping Luo[3], and Huchuan Lu[1,4]

[1] School of Information and Communication Engineering, Dalian University of
Technology, China [2] ByteDance [3] The University of Hong Kong
[4] Peng Cheng Laboratory

**Abstract.** We present a unified method, termed Unicorn, that can simultaneously solve four tracking problems (SOT, MOT, VOS, MOTS) with a single network using the same model parameters. Due to the fragmented definitions of the object tracking problem itself, most existing trackers are developed to address a single or part of tasks and overspecialize on the characteristics of specific tasks. By contrast, Unicorn provides a unified solution, adopting the same input, backbone, embedding, and head across all tracking tasks. For the first time, we accomplish the great unification of the tracking network architecture and learning paradigm. Unicorn performs on-par or better than its task-specific counterparts in 8 tracking datasets, including LaSOT, TrackingNet, MOT17, BDD100K, DAVIS16-17, MOTS20, and BDD100K MOTS. We believe that Unicorn will serve as a solid step towards the general vision model. Code is available at https://github.com/MasterBin-IIAU/Unicorn.

**Keywords:** object tracking

## 1 Introduction

Compared with weak AI designed for solving one specific task, artificial general intelligence (AGI) is expected to understand or learn any intellectual task that a human being can. Although there is still a large gap between this ambitious goal and the intellectual algorithms of today, some recent works [19,48,76,20] have begun to explore the possibility of building general vision models to address several vision tasks simultaneously.

Object tracking is one of the fundamental tasks in computer vision, which aims to build pixel-level or instance-level correspondence between frames and to output trajectories typically in the forms of boxes or masks. Over the years, according to different application scenarios, the object tracking problem has been mainly divided into four separate sub-tasks: Single Object Tracking (SOT) [17,39],

---

Multiple Object Tracking (MOT) [37,75], Video Object Segmentation (VOS) [43], and Multi-Object Tracking and Segmentation (MOTS) [57,75]. As a result, most tracking approaches are developed for only one of or part of the sub-tasks. Despite convenience for specific applications, this fragmented situation brings into the following drawbacks: (1) Trackers may over-specialize on the characteristic of specific sub-tasks, lacking in the generalization ability. (2) Independent model designs cause redundant parameters. For example, recent deep-learning-based trackers usually adopt similar backbones architectures, but the separate design philosophy hinders the potential reuse of parameters. It is natural to ask a question: Can all main-stream tracking tasks be solved by a unified model?

Although some works [60,33,58,66,36] attempt to unify SOT&VOS or MOT&MOTS by adding a mask branch to the existing box-level tracking system, there is still little progress towards the unification of SOT and MOT. There are mainly three obstacles hindering this process. (1) The characteristics of tracked objects vary. MOT usually tracks tens even hundreds of instances of specific categories. In contrast, SOT needs to track one target given in the reference frame no matter what class it belongs to. (2) SOT and MOT require different types of correspondence. SOT requires distinguishing the target from the background. However, MOT needs to match the currently detected objects with previous trajectories. (3) Most SOT methods [3,27,15,5,9,72] only take a small search region as the input to save computation and filter potential distractors. However, MOT algorithms [2,69,8,63,79,84,36] usually take the high-resolution full image as the input for detecting instances as completely as possible.

To conquer these challenges, we propose two core designs: the target prior and the pixel-wise correspondence. To be specific, (1) the target prior is an additional input for the detection head and serves as the switch among four tasks. For SOT&VOS, the target prior is the propagated reference target map, enabling the head to focus on the tracked target. For MOT&MOTS, by setting the target prior as zero, the head degenerates into the usual class-specific detection head smoothly. (2) The pixel-wise correspondence is the similarity between all pairs of points from the reference frame and the current frame. Both the SOT correspondence ($\mathbf{C}^{\text{SOT}} \in \mathbb{R}^{h'w' \times hw}$) and the MOT correspondence ($\mathbf{C}^{\text{MOT}} \in \mathbb{R}^{M \times N}$) are subsets of the pixel-wise correspondence ($\mathbf{C}_{\text{pix}} \in \mathbb{R}^{hw \times hw}$). (3) With the help of the informative target prior and the accurate pixel-wise correspondence, the design of the search region becomes unnecessary for SOT, leading to unified inputs as the full image for SOT and MOT.

Towards the **unif**ication of **o**bject t**r**ac**ki**ng, we propose Unicorn, a single network architecture to solve four tracking tasks. It takes the reference frame and the current frame as the inputs and produces their visual features by a weight-shared backbone. Then a feature interaction module is exploited to build pixel-wise correspondence between two frames. Based on the correspondence, a target prior is generated by propagating the reference target to the current frame. Finally, the target prior and the visual features are fused and sent to the detection head to get the tracked objects for all tasks.

With the unified network architecture, Unicorn can learn from various sources of tracking data and address four tracking tasks with the same model parameters. Extensive experiments show that Unicorn performs on-par or better than task-specific counterparts on 8 challenging benchmarks from four tracking tasks.

We summarize that our work has the following contributions:

– For the first time, Unicorn accomplishes the great unification of the network architecture and the learning paradigm for four tracking tasks.
– Unicorn bridges the gap among methods of four tracking tasks by the target prior and the pixel-wise correspondence.
– Unicorn puts forwards new state-of-the-art performance on 8 challenging tracking benchmarks with the same model parameters. This achievement will serve as a solid step towards the general vision model.

## 2  Related Work

### 2.1  Task-specific Trackers

SOT typically specifies one tracked target with a bounding box on the first frame, then requires trackers to predict boxes for the tracked target in the following frames. Considering the uniqueness and the motion continuity of the tracked target, most of the algorithms in SOT [3,27,70,15,5,9,72] track on a small search region rather than the whole image to reduce computation and to filter distractors. Although achieving great success in the SOT field, search-region-based trackers suffer from the following drawbacks: (1) Due to the limited visual field, it is difficult for these methods to recover from temporary tracking failure, especially in the long-term tracking scenarios. (2) The speed of these methods drops drastically as the number of tracked instances increases. The inefficiency problem restricts the application of SOT trackers in scenarios such as MOT, where there are tens or hundreds of targets to track. To overcome the first problem, some works [23,58] propose a global-detection-based tracking paradigm. However, these methods either require large modifications to the original detection architecture to integrate the target information or rely on complicated dynamic programming to pick the best tracklet. Besides, both Global-Track [23] and Siam R-CNN [23] are developed on two-stage Faster R-CNN, whose detection pipeline is tedious and relies on hand-crafted anchors and ROI-Align. By contrast, in this work, we build our method based on a one-stage, anchor-free detector [18]. Furthermore, we demonstrate that only with minimal change to the original detector architecture, we could transform an object detector into a powerful SOT tracker.

Different from SOT, MOT does not have any given prior on the first frame. Trackers of MOT are required to find and associate all instances of specific classes by themselves. The mainstream methods [63,79,84,41,49] follow the tracking-by-detection paradigm. Specifically, an MOT system typically has two main components, an object detector and a certain association strategy. Commonly used detectors include Faster R-CNN [45], the YOLO series [44,18], Center-Net [85], Sparse R-CNN[50], and Deformable DETR [88], etc. Popular associ-

ation methods include IoU matching [4,49], Kalman Filter [4,63,79], ReID embedding [65,41,63,79], Transformer [49,36,77], or the combination of them[78]. Although there are some works [87,12] introducing SOT trackers for the association, these SOT trackers [14,3] are completely independent with the MOT networks, without any weight sharing. There is still a large gap between methods of SOT and MOT.

The goal of VOS is to predict masks for the tracked instances based on the high-quality mask annotations of the first frame. This field is now dominated by memory-network-based methods [40,74,10]. Although achieving great performance, these methods suffer from the following disadvantages: (1) The memory network brings huge time and space complexity, especially when dealing with high spatial resolution and the long sequence. While these scenarios are quite common in sequences of SOT and MOT. Specifically, the long-term tracking benchmarks [17,55] in SOT usually have thousands of frames per sequence, being more than 20x longer than DAVIS [43]. Meanwhile, the image size in MOT [75] can reach 720x1280, while the image size of DAVIS is usually only 480x854. (2) SOTA methods assume that there are always high-quality mask annotations on the first frame. However, high-quality masks demand expensive labor costs and are usually unavailable in real-world applications. To overcome this problem, some works [60,33,58] attempt to develop weakly-annotated VOS algorithms, which only require box annotation on the first frame.

MOTS is highly related to MOT by changing the form of boxes to fine-grained representation of masks. MOTS benchmarks [57,75] are typically from the same scenarios as those of MOT [37,75]. Besides, many MOTS methods are developed upon MOT trackers. Representative approaches include 3D-convolution-based Track R-CNN [57] and Stem-Seg [1], Transformer-based TrackFormer [36], tracking-assisting-detection Trades [66] and Prototype-based PCAN [26].

## 2.2 General Vision Models

Despite the great success of specialized models for diverse tasks, there is still a large gap between the current AI with human-like, omnipotent Artificial General Intelligence (AGI). An important step towards this grand goal is to build a generalist model supporting a broad range of AI tasks. Recent pioneering works [19,48,76,20] attempt to approach this goal from different perspectives. Specifically, MuST [19] introduces a multi-task self-training pipeline, which harnesses the knowledge in independent specialized teacher models to train a single general student model. INTERN [48] proposes a new learning paradigm, which learns with supervisory signals from multiple sources in multiple stages. The developed general vision model generalizes well to different tasks but also has lower requirements on downstream data. Florence [76] is a new computer vision foundation model, which expands the representations to different tasks along space, time, and modality. Florence has great transferability and achieves new SOTA results on a wide range of vision benchmarks. OMNIVORE [20] proposes a modality-agnostic model which can classify images, videos, and single-view 3D data using the same model parameters.
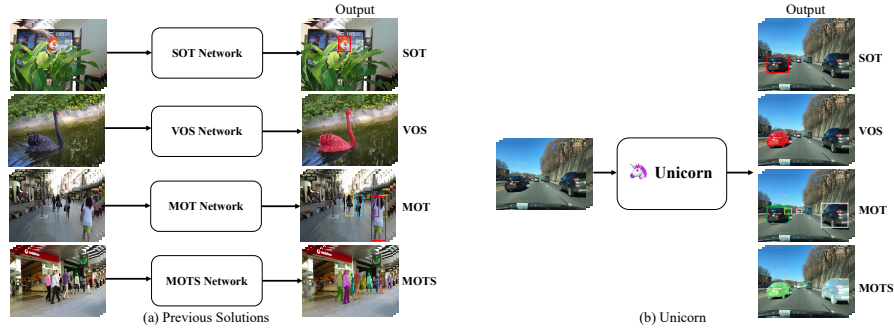
Fig. 1: Comparison between previous solutions and Unicorn.

### 2.3  Unification in Object Tracking

In the literature, some works [60,66,62] attempted to design a unified framework for supporting multiple tracking tasks. Specifically, SiamMask [60] is the first work to address SOT and VOS simultaneously. Similarly, TraDes [66] can solve both MOT and MOTS by introducing an extra mask head. Besides, UniTrack [62] proposes a high-level tracking framework, which consists of a shared appearance model and a series of unshared tracking heads. It demonstrates that different tracking tasks can share one appearance model for either propagation or association. However, the large discrepancy in tracking heads hinders it from exploiting a large amount of tracking data. Consequently, its performance lags far behind that of SOTA task-specific methods. Moreover, when used for MOT or MOTS, UniTrack requires extra, independent object detectors to provide observation. The extra object detector and the appearance model do not share the same backbone, bringing heavy burdens in parameters. By contrast, Unicorn solves four tracking tasks with one unified network with the same parameters. Besides, Unicorn can learn powerful representation from a large amount of labeled tracking data, achieving superior performance on 8 challenging benchmarks. Figure 1 shows the comparison between task-specific methods and Unicorn.

### 2.4  Correspondence Learning

Learning accurate correspondence is the key to many vision tasks, such as optical flow [51], video object segmentation [80,25], geometric matching [53,54], etc. The dense correspondence is usually obtained by computing correlation between the embedding maps of two frames. Most existing methods [51,80,25] obtain the embedding maps without considering the information exchange between two images. This could lead to ambiguous or wrong matching when there are many similar patterns or instances on the input images. Although some works [53,54] attempt to relieve this problem, they usually require complex optimization or uncertainty modeling. Different from the local comparison, Transformer [56] and
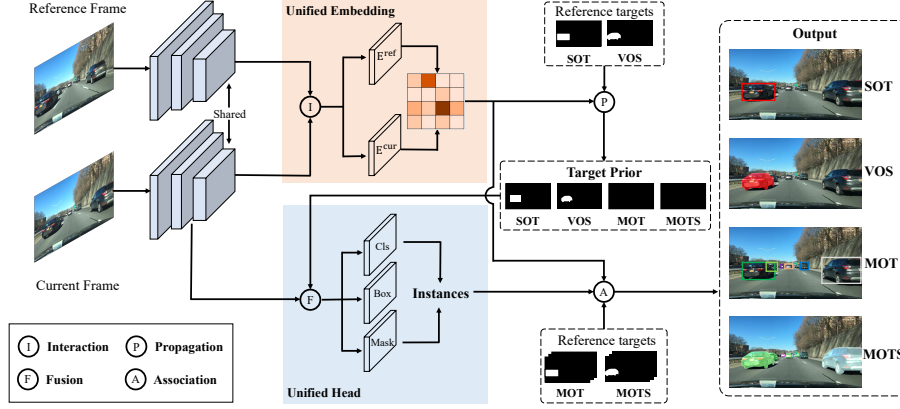
Fig. 2: Unicorn consists of three main components: (1) Unified inputs and backbone (2) Unified embedding (3) Unified head.

its variants [88] exploit the attention mechanism to capture the long-range dependency within the input sequence. In this work, we demonstrate that these operations can help to learn precise correspondence in object tracking.

## 3  Approach

We propose a unified solution for object tracking, called Unicorn, which consists of three main components: unified inputs and backbone; unified embedding and unified head. Three components are responsible for obtaining powerful visual representation, building precise correspondence and detecting diverse tracked targets respectively. The framework of Unicorn is demonstrated in Figure 2. Given the reference frame $\mathbf{I}_{ref}$, the current frame $\mathbf{I}_{cur}$, and the reference targets, Unicorn aims at predicting the states of the tracked targets on the current frame for four tasks with a unified network.

### 3.1  Unified Inputs and Backbone

For efficiently localizing multiple potential targets, Unicorn takes the whole image (for both the reference frame and the current frame) instead of local search regions as the inputs. This also endows Unicorn high resistance to tracking failure and the ability to re-detect tracked target after disappearance.

During the feature extraction, the reference frame and the current frame are passed through a weight-sharing backbone to get feature pyramid representations(FPN) [30]. To maintain important details and reduce the computational burden during computing correspondence, we choose the feature map with stride 16 as the input of the following embedding module. The corresponding features from the reference and the current frame are termed $\mathbf{F}_{ref}$ and $\mathbf{F}_{cur}$ respectively.

### 3.2   Unified Embedding

The core task of object tracking is to build accurate correspondence between frames in a video. For SOT and VOS, pixel-wise correspondence propagates the user-provided target from the reference frame (usually the $1^{st}$ frame) to the $t^{th}$ frame, providing strong prior information for the final box or mask prediction. Besides, for MOT and MOTS, instance-level correspondence helps to associate the detected instances on the $t^{th}$ frame to the existing trajectories on the reference frame (usually the $t - 1^{th}$ frame).

In Unicorn, given the spatially flattened reference frame embedding $\mathbf{E}_{\mathrm{ref}} \in \mathbb{R}^{hw \times c}$ and the current frame embedding $\mathbf{E}_{\mathrm{cur}} \in \mathbb{R}^{hw \times c}$, pixel-wise correspondence $\mathbf{C}_{\mathrm{pix}} \in \mathbb{R}^{hw \times hw}$ is computed by the matrix multiplication between them. For SOT&VOS taking the full image as the inputs, the correspondence is the the pixel-wise correspondence itself. For MOT&MOTS, assume that there are $M$ trajectories on the reference frame and $N$ detected instances on the current frame respectively, the instance-level correspondence $\mathbf{C}_{\mathrm{inst}} \in \mathbb{R}^{N \times M}$ is the matrix multiplication of the reference instance embedding $\mathbf{e}_{\mathrm{ref}} \in \mathbb{R}^{M \times c}$ and the current instance embedding $\mathbf{e}_{\mathrm{cur}} \in \mathbb{R}^{N \times c}$. The instance embedding $\mathbf{e}$ is extracted from the frame embedding $\mathbf{E}$, where the center of the instance is located.

$$
\begin{aligned}
\mathbf{C}_{\mathrm{pix}} &= \mathrm{softmax}(\mathbf{E}_{\mathrm{cur}} \mathbf{E}_{\mathrm{ref}}{}^{T}) \\
\mathbf{C}_{\mathrm{inst}} &= \mathrm{softmax}(\mathbf{e}_{\mathrm{cur}} \mathbf{e}_{\mathrm{ref}}{}^{T})
\end{aligned}
\tag{1}
$$

It can be seen that the instance-level correspondence $\mathbf{C}_{\mathrm{inst}}$ required by MOT and MOTS is the sub-matrix of the pixel-wise correspondence $\mathbf{C}_{\mathrm{pix}}$. Besides, learning highly discriminative embedding $\{\mathbf{E}_{\mathrm{ref}}, \mathbf{E}_{\mathrm{cur}}\}$ is the key to building precise correspondence for all tracking tasks.

**Feature Interaction.** Due to its advantages of capturing long-range dependency, Transformer [56] is an intuitive choice to enhance the original feature representation $\{\mathbf{F}_{\mathrm{ref}}, \mathbf{F}_{\mathrm{cur}}\}$. However, this could lead to huge memory cost when dealing with high-resolution feature maps, because the memory consumption increases with the length of the input sequence quadratically. To alleviate this problem, we replace the full attention with more memory-efficient deformable attention [88]. For more accurate correspondence, the enhanced feature maps are upsampled by $2\times$ to obtain high-resolution embeddings on the stride of 8.

$$
\{\mathbf{E}_{\mathrm{ref}}, \mathbf{E}_{\mathrm{cur}}\} = \mathrm{Upsample}(\mathrm{Attention}(\mathbf{F}_{\mathrm{ref}}, \mathbf{F}_{\mathrm{cur}}))
\tag{2}
$$

**Loss.** Ideal embedding should work well on both propagation(SOT, VOS) and association(MOT, MOTS). For SOT&VOS, although there is no human-annotated label for dense correspondence between frames, the embedding can be supervised by the difference between the propagated result $\widetilde{\mathbf{T}}_{\mathrm{cur}}$ and the ground-truth target map $\mathbf{T}_{\mathrm{cur}}$. Specifically, the shape of target map $\mathbf{T}$ is $hw \times 1$. The regions where the tracked target exists are equal to one and the other regions are equal

to zero. During the propagation, the pixel-wise correspondence $\mathbf{C}_{\text{pix}}$ transforms the reference target map $\mathbf{T}_{\text{ref}}$ to the estimation of the current target map $\widetilde{\mathbf{T}}_{\text{cur}}$.

$$\widetilde{\mathbf{T}}_{\text{cur}}(i,j) = \sum_k \mathbf{C}_{\text{pix}}(i,k) \cdot \mathbf{T}_{\text{ref}}(k,j) \tag{3}$$

Besides, for MOT and MOTS, the instance-level correspondence can be learned with standard contrastive learning paradigm. Specifically, assume that the instance $i$ from the current frame is matched with the instance $j$ from the reference frame, then the corresponding ground-truth matrix $\mathbf{G}$ should satisfies that

$$\mathbf{G}_{i,k} = \begin{cases} 0 & k \neq j \\ 1 & k = j \end{cases} \tag{4}$$

Finally, the unified embedding can be optimized end-to-end by Dice Loss [38] for SOT&VOS or Cross-Entropy Loss for MOT&MOTS.

$$\mathbf{L}_{\text{corr}} = \begin{cases} \text{Dice}(\widetilde{\mathbf{T}}_{\text{cur}}, \mathbf{T}_{\text{cur}}) & \text{task in } \{\text{SOT}, \text{VOS}\} \\ \text{CrossEntropy}(\mathbf{C}_{\text{inst}}, \mathbf{G}) & \text{task in } \{\text{MOT}, \text{MOTS}\} \end{cases} \tag{5}$$

### 3.3   Unified Head

To achieve the grand unification of object tracking, another important and challenging problem is designing a unified head for four tracking tasks. Specifically, MOT shall detect objects of specific categories. However, SOT needs to detect any target given in the reference frame. To bridge this gap, Unicorn introduces an extra input (called target prior) to the original detector head [18,52]. Without any further modification, Unicorn can easily detect various objects needed for four tasks with this unified head. More details about the head architecture can be found in the supplementary materials.

**Target Prior.** As mentioned in Sec. 3.2, given the reference target map $\mathbf{T}_{\text{ref}}$, the propagated target map $\widetilde{\mathbf{T}}_{\text{cur}}$ can provide strong prior information about the state of the tracked target. This motivates us to take it as a target prior when detecting targets for SOT&VOS. To be compatible with the original input of the detection head, we first reshape it to $h \times w \times 1$ (i.e. $\widetilde{\mathbf{T}}_{\text{cur}}^{\text{reshape}} \in \mathbb{R}^{h \times w \times 1}$). Meanwhile, when dealing with MOT&MOTS, we can simply set this prior to zero. Formally, the target prior $\mathbf{P}$ satisfies that

$$\mathbf{P} = \begin{cases} \widetilde{\mathbf{T}}_{\text{cur}}^{\text{reshape}} & \text{task in } \{\text{SOT}, \text{VOS}\} \\ \mathbf{0} & \text{task in } \{\text{MOT}, \text{MOTS}\} \end{cases} \tag{6}$$

**Feature Fusion.** The unified head takes the original FPN feature $\mathbf{F} \in \mathbb{R}^{h \times w \times c}$ and the target prior $\mathbf{P} \in \mathbb{R}^{h \times w \times 1}$ as the inputs. Unicorn fuses these two inputs with broadcast sum and passes the fused feature $\mathbf{F}' \in \mathbb{R}^{h \times w \times c}$ to the original detection head. This fusion strategy has the following advantages. (1)

The fused features are seamlessly compatible with four tasks. Specifically, for MOT&MOTS, the target prior is equal to zero. Then the fused feature $\mathbf{F}'$ degenerates back to the original FPN feature $\mathbf{F}$ to detect objects of specific classes. For SOT&VOS, the target prior with strong target information can enhance the original FPN feature and makes the network focus on the tracked target.(2) The architecture is simple, without introducing complex changes to the original detection head. Furthermore, the consistent architecture also enables Unicorn to fully exploit the pretrained weights of the original object detector.

### 3.4   Training and Inference

**Training.** The whole training process divides into two stages: SOT-MOT joint training and VOS-MOTS joint training. In the first stage, the network is end-to-end optimized with the correspondence loss and the detection loss using data from SOT&MOT. In the second stage, a mask branch is added and optimized with the mask loss using data from VOS&MOTS with other parameters fixed.

**Inference.** During the test phase, for SOT&VOS, the reference target map is generated once on the first frame and kept fixed in the following frames. Unicorn directly picks the box or mask with the highest confidence score as the final tracking result, without any hyperparameter-sensitive post-processing like cosine window. Besides, Unicorn only needs to run the heavy backbone and the correspondence once, while running the lightweight head rather than the whole network $N$ times, leading to higher efficiency. For MOT&MOTS, Unicorn detects all objects of the given categories and simultaneously outputs corresponding instance embeddings. The later association is performed based on the embeddings and the motion model for BDD100K and MOT17 respectively.

## 4   Experiments

### 4.1   Implementation Details

When comparing with state-of-the-art methods, we choose ConvNeXt-Large [31] as the backbone. In ablations, we report the results of our method with ConvNeXt-Tiny [31] and ResNet-50 [21] as the backbone. The input image size is $800 \times 1280$ and the shortest side ranges from 736 to 864 during multi-scale training. The model is trained on 16 NVIDIA Tesla A100 GPU with a global batch size of 32. To avoid inaccurate statistics estimation, we replace all Batch Normalization [24] with Group Normalization [67]. Two training stages randomly sample data from SOT&MOT datasets and VOS&MOTS datasets, respectively. Each training stage consists of 15 epochs with 200,000 pairs of frames in every epoch. The optimizer is Adam-W [32] with weight decay of $5e^{-4}$ and momentum of 0.9. The initial learning rate is $2.5e^{-4}$ with 1 epoch warm-up and the cosine annealing schedule. More details can be found in the supplementary materials. In

Table 1: State-of-the-art comparison on LaSOT [17] and TrackingNet [39].

| Method | Source | LaSOT [17] | | | TrackingNet [39] | | |
|---|---|---|---|---|---|---|---|
| | | Success | $P_{norm}$ | P | Success | $P_{norm}$ | P |
| SiamFC [3] | ECCVW2016 | 33.6 | 42.0 | 33.9 | 57.1 | 66.3 | 53.3 |
| UniTrack [62] | NeurIPS2021 | 35.1 | - | 32.6 | - | - | - |
| ATOM [15] | CVPR2019 | 51.5 | 57.6 | 50.5 | 70.3 | 77.1 | 64.8 |
| SiamPRN++ [27] | CVPR2019 | 49.6 | 56.9 | 49.1 | 73.3 | 80.0 | 69.4 |
| DiMP [5] | ICCV2019 | 56.9 | 65.0 | 56.7 | 74.0 | 80.1 | 68.7 |
| GlobalTrack [23] | AAAI2020 | 52.1 | - | 52.7 | 70.4 | 75.4 | 65.6 |
| SiamFC++ [70] | AAAI2020 | 54.4 | 62.3 | 54.7 | 75.4 | 80.0 | 70.5 |
| D3S [33] | CVPR2020 | - | - | - | 72.8 | 76.8 | 66.4 |
| PrDiMP [16] | CVPR2020 | 59.8 | 68.8 | 60.8 | 75.8 | 81.6 | 70.4 |
| Siam R-CNN [58] | CVPR2020 | 64.8 | 72.2 | - | 81.2 | 85.4 | 80.0 |
| KYS [6] | ECCV2020 | 55.4 | 63.3 | - | 74.0 | 80.0 | 68.8 |
| Ocean [82] | ECCV2020 | 56.0 | 65.1 | 56.6 | - | - | - |
| TrDiMP [59] | CVPR2021 | 63.9 | - | 61.4 | 78.4 | 83.3 | 73.1 |
| TransT [9] | CVPR2021 | 64.9 | 73.8 | 69.0 | 81.4 | 86.7 | 80.3 |
| AutoMatch [81] | ICCV2021 | 58.2 | - | 59.9 | 76.0 | - | 72.6 |
| SAOT [86] | ICCV2021 | 61.6 | 70.8 | - | - | - | - |
| KeepTrack [35] | ICCV2021 | 67.1 | **77.2** | 70.2 | - | - | - |
| STARK [72] | ICCV2021 | 67.1 | 77.0 | - | 82.0 | **86.9** | - |
| **Unicorn** | Ours | **68.5** | 76.6 | **74.1** | **83.0** | 86.4 | **82.2** |

Section 4.2-4.5, we compare Unicorn with task-specific counterparts in 8 tracking datasets. In each benchmark, the **red bold** font and the blue font indicate the best two results. Unicorn in four tasks uses the same model parameters.

## 4.2   Evaluations on Single Object Tracking

We compare Unicorn with state-of-the-art SOT trackers on two popular and challenging benchmarks, LaSOT [17] and TrackingNet [39]. Both datasets evaluate the tracking performance with the following measures: Success, precision($P$) and normalized precision($P_{norm}$). All these measures are the higher the better.

**LaSOT**. LaSOT [17] is a large-scale long-term tracking benchmark, which contains 280 videos in the test set with an average length of 2448 frames. Tab 1 shows that Unicorn achieves new state-of-the-art Success and Precision of 68.5% and 74.1% respectively. It is also worth noting that Unicorn surpasses the previous best global-detection-based tracker Siam R-CNN [58] by a large margin (68.5% vs 64.8%) with a much simpler network architecture and tracking strategy (directly picking the top-1 vs tracklet dynamic programming).

**TrackingNet**. TrackingNet [39] is a large-scale short-term tracking benchmark containing 511 videos in the test set. As reported in Tab 1, Unicorn surpasses all previous methods with a Success of 83.0% and a Precision of 82.2%.

Table 2: State-of-the-art comparison on MOT17 [37] test set.

| Tracker | MOTA↑ | IDF1↑ | HOTA↑ | MT↑ | ML↓ | FP↓ | FN↓ | IDs↓ |
|---|---|---|---|---|---|---|---|---|
| TrackFormer [36] | 65.0 | 63.9 | - | - | - | 70443 | 123552 | 3528 |
| Chained-Tracker [42] | 66.6 | 57.4 | 49.0 | 37.8% | 18.5% | 22284 | 160491 | 5529 |
| CenterTrack [84] | 67.8 | 64.7 | 52.2 | 34.6% | 24.6% | 18498 | 160332 | 3039 |
| QuasiDense [41] | 68.7 | 66.3 | 53.9 | 40.6% | 21.9% | 26589 | 146643 | 3378 |
| TraDes [66] | 69.1 | 63.9 | 52.7 | 36.4% | 21.5% | 20892 | 150060 | 3555 |
| SOTMOT [83] | 71.0 | 71.9 | - | 42.7% | 15.3% | 39537 | 118983 | 5184 |
| TransCenter [68] | 73.2 | 62.2 | 54.5 | 40.8% | 18.5% | 23112 | 123738 | 4614 |
| MOTR [77] | 73.4 | 68.6 | 57.8 | 42.9% | 19.1% | 27939 | 119589 | 2439 |
| FairMOT [79] | 73.7 | 72.3 | 59.3 | 43.2% | 17.3% | 27507 | 117477 | 3303 |
| CSTrack [28] | 74.9 | 72.6 | 59.3 | 41.5% | 17.5% | 23847 | 114303 | 3567 |
| TransTrack [49] | 75.2 | 63.5 | 54.1 | 55.3% | 10.2% | 50157 | 86442 | 3603 |
| OMC [29] | 76.3 | 72.3 | - | 44.8% | 15.5% | - | - | - |
| CorrTracker [61] | 76.5 | 73.6 | 60.7 | 47.6% | 12.7% | 29808 | 99510 | 3369 |
| TransMOT [13] | 76.7 | 75.1 | 61.7 | 51.0% | 16.4% | 36231 | 93150 | 2346 |
| **Unicorn** | **77.2** | **75.5** | **61.7** | **58.7%** | 11.2% | 50087 | **73349** | 5379 |

Table 3: State-of-the-art comparison on BDD100K [75] tracking validation set.

| Method | Split | mMOTA↑ | mIDF1↑ | MOTA↑ | IDF1↑ | FN↓ | FP↓ | ID Sw.↓ | MT↑ | ML↓ | mAP↑ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Yu *et al.* [75] | val | 25.9 | 44.5 | 56.9 | 66.8 | 122406 | 52372 | 8315 | 8396 | 3795 | 28.1 |
| QDTrack [41] | val | 36.6 | 50.8 | 63.5 | **71.5** | 108614 | 46621 | **6262** | 9481 | 3034 | 32.6 |
| **Unicorn** | val | **41.2** | **54.0** | **66.6** | 71.3 | **95454** | **41648** | 10876 | **10296** | **2505** | **41.4** |

## 4.3 Evaluations on Multiple Object Tracking

We compare Unicorn with state-of-the-art MOT trackers on two challenging benchmarks: MOT17 [37] and BDD100K [75]. The common metrics include Multiple-Object Tracking Accuracy (MOTA), Identity F1 Score (IDF1), False Positives (FP), False Negatives (FN), the percentage of Mostly Tracked Trajectories (MT) and Mostly Lost Trajectories (ML), Identity Switches (IDS). Among them, MOTA is the primary metric to measure the overall detection and tracking performance, IDF1 is used to measure the trajectory identity accuracy.

**MOT17**. The MOT17 focuses on pedestrian tracking and includes 7 sequences in the training set and 7 sequences in the test set. We compare Unicorn with previous methods under the private detection protocol on the test set of MOT17. Tab 2 demonstrates that Unicorn achieves the best MOTA and IDF1, surpassing the previous SOTA method by 0.5% and 0.4% respectively.

**BDD100K MOT**. BDD100K is a large-scale dataset of visual driving scenes and requires tracking 8 categories of instances. To evaluate the average performance across 8 classes, BDD100K additionally introduces two measures: mMOTA and mIDF1. Different from MOT17, BDD100K is annotated at only 5 FPS. The low frame-rate brings difficulty to motion models commonly used for MOT17.

Table 4: State-of-the-art comparison on the validation set of the DAVIS-2016 and the DAVIS-2017. OL: online learning, Memory: using an external memory bank.

| Init | Method | OL | Memory | $(\mathcal{J}\&\mathcal{F})^{16}$ | $\mathcal{J}^{16}$ | $\mathcal{F}^{16}$ | $(\mathcal{J}\&\mathcal{F})^{17}$ | $\mathcal{J}^{17}$ | $\mathcal{F}^{17}$ |
|---|---|---|---|---|---|---|---|---|---|
| mask | FAVOS [11] | ✗ | ✗ | 81.0 | 82.4 | 79.5 | 58.2 | 54.6 | 61.8 |
| | OSMN [34] | ✗ | ✗ | 73.5 | 74.0 | 72.9 | 54.8 | 52.5 | 57.1 |
| | VideoMatch [22] | ✗ | ✗ | - | 81.0 | - | 56.5 | - | - |
| | UniTrack [62] | ✗ | ✓ | - | - | - | - | 58.4 | - |
| | RANet [64] | ✗ | ✗ | 85.5 | 85.5 | 85.4 | 65.7 | 63.2 | 68.2 |
| | FRTM [46] | ✓ | ✓ | 83.5 | 83.6 | 83.4 | 76.7 | 73.9 | 79.6 |
| | TVOS [80] | ✗ | ✓ | - | - | - | 72.3 | 69.9 | 74.7 |
| | LWL [7] | ✓ | ✓ | - | - | - | 81.6 | 79.1 | 84.1 |
| | STM [40] | ✗ | ✓ | 89.3 | 88.7 | 89.9 | 81.8 | 79.2 | 84.3 |
| | CFBI [74] | ✗ | ✓ | 89.4 | 88.3 | 90.5 | 81.9 | 79.1 | 84.6 |
| | HMMN [47] | ✗ | ✓ | 90.8 | 89.6 | 92.0 | 84.7 | 81.9 | 87.5 |
| | STCN [10] | ✗ | ✓ | 91.6 | 90.8 | 92.5 | 85.4 | 82.2 | 88.6 |
| bbox | SiamMask [60] | ✗ | ✗ | 69.8 | 71.7 | 67.8 | 56.4 | 54.3 | 58.5 |
| | D3S [33] | ✗ | ✗ | 74.0 | 75.4 | 72.6 | 60.8 | 57.8 | 63.8 |
| | Siam R-CNN [58] | ✗ | ✗ | - | - | - | 70.6 | 66.1 | 75.0 |
| | **Unicorn** | ✗ | ✗ | **87.4** | **86.5** | **88.2** | 69.2 | 65.2 | 73.2 |

As shown in Tab 3, Unicorn achieves the best performance, largely surpassing the previous SOTA method QDTrack [41] on the val set. Specifically, the improvement is up to 4.6% and 3.2% in terms of mMOTA and mIDF1 respectively.

### 4.4    Evaluations on Video Object Segmentation

We further evaluate the ability of Unicorn to perform VOS on DAVIS [43] 2016 and 2017. Both datasets evaluate methods with the region similarity $\mathcal{J}$, the contour accuracy $\mathcal{F}$, and the average of them $\mathcal{J}\&\mathcal{F}$.

**DAVIS-16**. DAVIS-16 includes 20 single-object videos in the validation set. Tab. 4 demonstrates that Unicorn achieves the best results among methods with bounding-box initialization, even surpassing RANet [64] and FRTM [46] with mask initialization. Meanwhile, Unicorn outperforms its multi-task counterparts SiamMask [60] by a large margin of 17.6% in terms of $\mathcal{J}\&\mathcal{F}$.

**DAVIS-17**. DAVIS-17 contains 30 videos in the validation set and there could be multiple tracked targets in each sequence. As shown in Tab. 4, compared with the previous best box-initialized method Siam R-CNN [58], Unicorn achieves competitive results with a much simpler architecture. Specifically, Siam R-CNN [58] uses an extra Box2Seg network, which is completely independent from the box-based tracker without any weight sharing. However, Unicorn can predict both boxes and masks with a unified head. Although there is still gap between the performance of Unicorn with that of SOTA VOS methods with mask initialization, Unicorn can address four tracking tasks with the same model parameters, while HMMN [47] and STCN [10] can only be used in the VOS task.

Table 5: State-of-the-art comparison on the MOTS [57] test set.

| Method | sMOTSA↑ | IDF1↑ | MT↑ | ML ↓ | FP↓ | FN↓ | ID Sw.↓ |
|---|---|---|---|---|---|---|---|
| Track R-CNN [57] | 40.6 | 42.4 | 38.7% | 21.6% | 1261 | 12641 | 567 |
| TraDeS [66] | 50.8 | 58.7 | 49.4% | 18.3% | 1474 | 9169 | 492 |
| TrackFormer [36] | 54.9 | 63.6 | - | - | 2233 | 7195 | **278** |
| PointTrackV2 [71] | 62.3 | 42.9 | 56.7% | 12.5% | **963** | 5993 | 541 |
| **Unicorn** | **65.3** | **65.9** | **64.9%** | **10.1%** | 1364 | **4452** | 398 |

Table 6: State-of-the-art comparison on the BDD100K MOTS validation set.

| Method | Online | mMOTSA↑ | mMOTSP↑ | mIDF1↑ | ID Sw.↓ | mAP↑ |
|---|---|---|---|---|---|---|
| SortIoU | ✓ | 10.3 | 59.9 | 21.8 | 15951 | 22.2 |
| MaskTrackRCNN [73] | ✓ | 12.3 | 59.9 | 26.2 | 9116 | 22.0 |
| STEm-Seg [1] | ✗ | 12.2 | 58.2 | 25.4 | 8732 | 21.8 |
| QDTrack-mots [41] | ✓ | 22.5 | 59.6 | 40.8 | 1340 | 22.4 |
| QDTrack-mots-fix [41] | ✓ | 23.5 | 66.3 | 44.5 | 973 | 25.5 |
| PCAN [26] | ✓ | 27.4 | 66.7 | **45.1** | **876** | 26.6 |
| **Unicorn** | ✓ | **29.6** | **67.7** | 44.2 | 1731 | **32.1** |

### 4.5 Evaluations on Multi-Object Tracking and Segmentation

Finally, we evaluate the ability of Unicorn for MOTS on MOTS20 [57] and BDD100K MOTS [75]. The main evaluation metrics are sMOTSA and mMOTSA.

**MOTS20 Challenge**. MOTS20 Challenge has 4 sequences in the test set. As shown in Tab. 5, Unicorn achieves state-of-the-art performance, surpassing the second-best method PoinTrackV2 [71] by a large margin of 3.3% on sMOTSA.

**BDD100K MOTS Challenge**. BDD100K MOTS Challenge includes 37 sequences in the validation set. Tab. 6 demonstrates that Unicorn outperforms the previous best method PCAN [26] by a large margin (i.e. mMOTSA +2.2%, mAP +5.5%). Meanwhile, Unicorn does not use any complex design like space-time memory or prototypical network as in PCAN, bringing into a simpler pipeline.

### 4.6 Ablations and the Other Analysis

For the ablations, we choose Unicorn with ConvNeXt-Tiny [31] backbone as the baseline. The detailed results are demonstrated in Tab. 7.

**Backbone**. We implement a variant of Unicorn with ResNet-50 [21] as the backbone. Although the overall performance of this version is lower than the baseline, this variant still achieves superior performance on four tasks.

**Interaction**. Besides the memory-efficient deformable attention [88], we compare the full attention [56] and the convolution operation, which does not exchange information between frames. Experiments show that deformable attention obtains better performance than the full attention, while consuming much

Table 7: Ablations and comparisons. Our baseline model are underlined.

| Experiment | Method | SOT LaSOT(AUC) | MOT BDD(mMOTA) | VOS DAVIS17($\mathcal{J}\&\mathcal{F}$) | MOTS BDD(mMOTSA) | FPS |
|---|---|---|---|---|---|---|
| Backbone | ConvNeXt-Tiny | 67.7 | 39.9 | 68.0 | 29.7 | 14 |
| | ResNet-50 | 65.3 | 35.1 | 66.2 | 30.8 | 13 |
| Interaction | Deformable Att | 67.7 | 39.9 | 68.0 | 29.7 | 14 |
| | Full Att | 67.1 | 38.5 | 66.9 | 26.7 | 13 |
| | Conv | 66.8 | 37.6 | 66.6 | 27.0 | 15 |
| Fusion | Broad Sum | 67.7 | 39.9 | 68.0 | 29.7 | 14 |
| | Concat | 66.8 | 38.3 | 66.7 | 27.2 | 14 |
| | W/o Prior | 50.9 | 37.6 | 29.2 | 27.8 | 14 |
| Single task | Unification | 67.7 | 39.9 | 68.0 | 29.7 | 14 |
| | SOT only | 67.5 | - | - | - | 14 |
| | MOT only | - | 39.6 | - | - | 14 |
| | VOS only | - | - | 68.4 | - | 14 |
| | MOTS only | - | - | - | 28.1 | 14 |
| Speed | Ours | 67.7 | 39.9 | 68.0 | 29.7 | 14 |
| | Ours-RT | 67.1 | 37.5 | 66.8 | 26.2 | 23 |

less memory. Moreover, the results of the convolution are lower than the baseline, showing the importance of interaction for accurate correspondence.

**Fusion**. Apart from broadcast sum, we compare other two methods: concatenation, and without the target prior. The performance of SOT and VOS drops significantly after removing the target prior, demonstrating the importance of this design. Besides, broadcast sum performs better than concatenation.

**Single Task**. We compare with training four independent models for different tasks. Experiments show that our unified model performs on-par with independently trained counterparts, while being much more parameter-efficient.

**Speed**. We develop a light-weight variant with a lower input resolution of 640x1024. Experiments show that the real-time version does not only achieves competitive performance but also can run in real-time at more than 20 FPS.

## 5    Conclusions

We propose Unicorn, a unified approach to address four tracking tasks using a single model with the same model parameters. For the first time, it achieves the unification of network architecture and learning paradigm for object tracking. Extensive experiments demonstrate that Unicorn performs on-par or better than task-specific counterparts on 8 challenging benchmarks. We hope that Unicorn can serve as a solid step towards the general vision model.

# References

1. Athar, A., Mahadevan, S., Osep, A., Leal-Taixé, L., Leibe, B.: STEm-Seg: Spatio-temporal embeddings for instance segmentation in videos. In: ECCV (2020) 4, 13

2. Bergmann, P., Meinhardt, T., Leal-Taixe, L.: Tracking without bells and whistles. In: ICCV (2019) 2

3. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.S.: Fully-convolutional siamese networks for object tracking. In: ECCVW (2016) 2, 3, 4, 11

4. Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple online and realtime tracking. In: ICIP (2016) 4

5. Bhat, G., Danelljan, M., Gool, L.V., Timofte, R.: Learning discriminative model prediction for tracking. In: ICCV (2019) 2, 3, 11

6. Bhat, G., Danelljan, M., Gool, L.V., Timofte, R.: Know your surroundings: Exploiting scene information for object tracking. In: ECCV (2020) 11

7. Bhat, G., Lawin, F.J., Danelljan, M., Robinson, A., Felsberg, M., Gool, L.V., Timofte, R.: Learning what to learn for video object segmentation. In: ECCV (2020) 12

8. Brasó, G., Leal-Taixé, L.: Learning a neural solver for multiple object tracking. In: CVPR (2020) 2

9. Chen, X., Yan, B., Zhu, J., Wang, D., Yang, X., Lu, H.: Transformer tracking. In: CVPR (2021) 2, 3, 11

10. Cheng, H.K., Tai, Y.W., Tang, C.K.: Rethinking space-time networks with improved memory coverage for efficient video object segmentation. NeurIPS (2021) 4, 12, 13

11. Cheng, J., Tsai, Y.H., Hung, W.C., Wang, S., Yang, M.H.: Fast and accurate online video object segmentation via tracking parts. In: CVPR (2018) 12

12. Chu, P., Ling, H.: FAMNet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking. In: ICCV (2019) 4

13. Chu, P., Wang, J., You, Q., Ling, H., Liu, Z.: TransMOT: Spatial-temporal graph transformer for multiple object tracking. arXiv preprint arXiv:2104.00194 (2021) 11

14. Danelljan, M., Bhat, G., Khan, F.S., Felsberg, M.: ECO: Efficient convolution operators for tracking. In: CVPR (2017) 4

15. Danelljan, M., Bhat, G., Khan, F.S., Felsberg, M.: ATOM: Accurate tracking by overlap maximization. In: CVPR (2019) 2, 3, 11

16. Danelljan, M., Gool, L.V., Timofte, R.: Probabilistic regression for visual tracking. In: CVPR (2020) 11

17. Fan, H., Lin, L., Yang, F., Chu, P., Deng, G., Yu, S., Bai, H., Xu, Y., Liao, C., Ling, H.: LaSOT: A high-quality benchmark for large-scale single object tracking. In: CVPR (2019) 1, 4, 10, 11

18. Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J.: YOLOX: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430 (2021) 3, 8

19. Ghiasi, G., Zoph, B., Cubuk, E.D., Le, Q.V., Lin, T.Y.: Multi-task self-training for learning general representations. In: ICCV (2021) 1, 4

20. Girdhar, R., Singh, M., Ravi, N., van der Maaten, L., Joulin, A., Misra, I.: Omnivore: A single model for many visual modalities. arXiv preprint arXiv:2201.08377 (2022) 1, 4

21. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016) 9, 13
22. Hu, Y.T., Huang, J.B., Schwing, A.G.: Videomatch: Matching based video object segmentation. In: ECCV (2018) 12
23. Huang, L., Zhao, X., Huang, K.: Globaltrack: A simple and strong baseline for long-term tracking. In: AAAI (2020) 3, 11
24. Ioffe, S., Szegedy, C.: Batch Normalization: Accelerating deep network training by reducing internal covariate shift. In: ICML (2015) 9
25. Jabri, A., Owens, A., Efros, A.: Space-time correspondence as a contrastive random walk. NeurIPS (2020) 5
26. Ke, L., Li, X., Danelljan, M., Tai, Y.W., Tang, C.K., Yu, F.: Prototypical cross-attention networks for multiple object tracking and segmentation. NeurIPS (2021) 4, 13
27. Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J., Yan, J.: SiamRPN++: Evolution of siamese visual tracking with very deep networks. In: CVPR (2019) 2, 3, 11
28. Liang, C., Zhang, Z., Lu, Y., Zhou, X., Li, B., Ye, X., Zou, J.: Rethinking the competition between detection and reid in multi-object tracking. arXiv preprint arXiv:2010.12138 (2020) 11
29. Liang, C., Zhang, Z., Zhou, X., Li, B., Lu, Y., Hu, W.: One more check: Making" fake background" be tracked again. arXiv preprint arXiv:2104.09441 (2021) 11
30. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR (2017) 6
31. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. arXiv preprint arXiv:2201.03545 (2022) 9, 13
32. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017) 10
33. Lukezic, A., Matas, J., Kristan, M.: D3S-a discriminative single shot segmentation tracker. In: CVPR (2020) 2, 4, 11, 12
34. Maninis, K.K., Caelles, S., Chen, Y., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., Van Gool, L.: Video object segmentation without temporal information. TPAMI (2018) 12
35. Mayer, C., Danelljan, M., Paudel, D.P., Van Gool, L.: Learning target candidate association to keep track of what not to track. In: ICCV (2021) 11
36. Meinhardt, T., Kirillov, A., Leal-Taixe, L., Feichtenhofer, C.: TrackFormer: Multi-object tracking with transformers. arXiv preprint arXiv:2101.02702 (2021) 2, 4, 11, 13
37. Milan, A., Leal-Taixé, L., Reid, I., Roth, S., Schindler, K.: MOT16: A benchmark for multi-object tracking. arXiv preprint arXiv:1603.00831 (2016) 2, 4, 10, 11
38. Milletari, F., Navab, N., Ahmadi, S.A.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 3DV (2016) 8
39. Muller, M., Bibi, A., Giancola, S., Alsubaihi, S., Ghanem, B.: Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In: ECCV (2018) 1, 10, 11
40. Oh, S.W., Lee, J.Y., Xu, N., Kim, S.J.: Video object segmentation using space-time memory networks. In: ICCV (2019) 4, 12
41. Pang, J., Qiu, L., Li, X., Chen, H., Li, Q., Darrell, T., Yu, F.: Quasi-dense similarity learning for multiple object tracking. In: CVPR (2021) 3, 4, 10, 11, 12, 13
42. Peng, J., Wang, C., Wan, F., Wu, Y., Wang, Y., Tai, Y., Wang, C., Li, J., Huang, F., Fu, Y.: Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking. In: ECCV (2020) 11

43. Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., Van Gool, L.: The 2017 davis challenge on video object segmentation. arXiv preprint arXiv:1704.00675 (2017) 2, 4, 12

44. Redmon, J., Farhadi, A.: YOLOv3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018) 3

45. Ren, S., He, K., Girshick, R., Sun, J.: Faster R–CNN: Towards real-time object detection with region proposal networks. In: NeurIPS (2015) 3

46. Robinson, A., Lawin, F.J., Danelljan, M., Khan, F.S., Felsberg, M.: Learning fast and robust target models for video object segmentation. In: CVPR (2020) 12

47. Seong, H., Oh, S.W., Lee, J.Y., Lee, S., Lee, S., Kim, E.: Hierarchical memory matching network for video object segmentation. In: ICCV (2021) 12, 13

48. Shao, J., Chen, S., Li, Y., Wang, K., Yin, Z., He, Y., Teng, J., Sun, Q., Gao, M., Liu, J., et al.: INTERN: A new learning paradigm towards general vision. arXiv preprint arXiv:2111.08687 (2021) 1, 4

49. Sun, P., Jiang, Y., Zhang, R., Xie, E., Cao, J., Hu, X., Kong, T., Yuan, Z., Wang, C., Luo, P.: TransTrack: Multiple-object tracking with transformer. arXiv preprint arXiv:2012.15460 (2020) 3, 4, 11

50. Sun, P., Zhang, R., Jiang, Y., Kong, T., Xu, C., Zhan, W., Tomizuka, M., Li, L., Yuan, Z., Wang, C., et al.: Sparse r-cnn: End-to-end object detection with learnable proposals. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14454–14463 (2021) 3

51. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: ECCV (2020) 5

52. Tian, Z., Shen, C., Chen, H.: Conditional convolutions for instance segmentation. In: ECCV (2020) 8

53. Truong, P., Danelljan, M., Gool, L.V., Timofte, R.: Gocor: Bringing globally optimized correspondence volumes into your neural network. NeurIPS (2020) 5

54. Truong, P., Danelljan, M., Van Gool, L., Timofte, R.: Learning accurate dense correspondences and when to trust them. In: CVPR (2021) 5

55. Valmadre, J., Bertinetto, L., Henriques, J.F., Tao, R., Vedaldi, A., Smeulders, A.W., Torr, P.H., Gavves, E.: Long-term tracking in the wild: a benchmark. In: ECCV (2018) 4

56. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: NeurIPS (2017) 6, 7, 14

57. Voigtlaender, P., Krause, M., Osep, A., Luiten, J., Sekar, B.B.G., Geiger, A., Leibe, B.: MOTS: Multi-object tracking and segmentation. In: CVPR (2019) 2, 4, 13

58. Voigtlaender, P., Luiten, J., Torr, P.H., Leibe, B.: Siam R-CNN: Visual tracking by re-detection. In: CVPR (2020) 2, 3, 4, 10, 11, 12

59. Wang, N., Zhou, W., Wang, J., Li, H.: Transformer meets tracker: Exploiting temporal context for robust visual tracking. In: CVPR (2021) 11

60. Wang, Q., Zhang, L., Bertinetto, L., Hu, W., Torr, P.H.S.: Fast online object tracking and segmentation: A unifying approach. In: CVPR (2019) 2, 4, 5, 12

61. Wang, Q., Zheng, Y., Pan, P., Xu, Y.: Multiple object tracking with correlation learning. In: CVPR (2021) 11

62. Wang, Z., Zhao, H., Li, Y.L., Wang, S., Torr, P., Bertinetto, L.: Do different tracking tasks require different appearance models? NeurIPS (2021) 5, 11, 12

63. Wang, Z., Zheng, L., Liu, Y., Li, Y., Wang, S.: Towards real-time multi-object tracking. In: ECCV (2020) 2, 3, 4

64. Wang, Z., Xu, J., Liu, L., Zhu, F., Shao, L.: Ranet: Ranking attention network for fast video object segmentation. In: ICCV (2019) 12

65. Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. In: ICIP (2017) 4

66. Wu, J., Cao, J., Song, L., Wang, Y., Yang, M., Yuan, J.: Track to detect and segment: An online multi-object tracker. In: CVPR (2021) 2, 4, 5, 11, 13

67. Wu, Y., He, K.: Group Normalization. In: ECCV (2018) 9

68. Xu, Y., Ban, Y., Delorme, G., Gan, C., Rus, D., Alameda-Pineda, X.: Transcenter: Transformers with dense queries for multiple-object tracking. arXiv preprint arXiv:2103.15145 (2021) 11

69. Xu, Y., Osep, A., Ban, Y., Horaud, R., Leal-Taixé, L., Alameda-Pineda, X.: How to train your deep multi-object tracker. In: CVPR (2020) 2

70. Xu, Y., Wang, Z., Li, Z., Yuan, Y., Yu, G.: SiamFC++: towards robust and accurate visual tracking with target estimation guidelines. In: AAAI (2020) 3, 11

71. Xu, Z., Yang, W., Zhang, W., Tan, X., Huang, H., Huang, L.: Segment as points for efficient and effective online multi-object tracking and segmentation. TPAMI (2021) 13

72. Yan, B., Peng, H., Fu, J., Wang, D., Lu, H.: Learning spatio-temporal transformer for visual tracking. In: ICCV (2021) 2, 3, 11

73. Yang, L., Fan, Y., Xu, N.: Video instance segmentation. In: ICCV (2019) 13

74. Yang, Z., Wei, Y., Yang, Y.: Collaborative video object segmentation by foreground-background integration. In: ECCV (2020) 4, 12

75. Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., Darrell, T.: BDD100K: A diverse driving dataset for heterogeneous multitask learning. In: CVPR (2020) 2, 4, 10, 12, 13

76. Yuan, L., Chen, D., Chen, Y.L., Codella, N., Dai, X., Gao, J., Hu, H., Huang, X., Li, B., Li, C., et al.: Florence: A new foundation model for computer vision. arXiv preprint arXiv:2111.11432 (2021) 1, 4

77. Zeng, F., Dong, B., Wang, T., Zhang, X., Wei, Y.: MOTR: End-to-end multiple-object tracking with transformer. arXiv preprint arXiv:2105.03247 (2021) 4, 11

78. Zhang, Y., Sun, P., Jiang, Y., Yu, D., Yuan, Z., Luo, P., Liu, W., Wang, X.: Bytetrack: Multi-object tracking by associating every detection box. arXiv preprint arXiv:2110.06864 (2021) 4

79. Zhang, Y., Wang, C., Wang, X., Zeng, W., Liu, W.: FairMOT: On the fairness of detection and re-identification in multiple object tracking. IJCV (2021) 2, 3, 4, 11

80. Zhang, Y., Wu, Z., Peng, H., Lin, S.: A transductive approach for video object segmentation. In: CVPR (2020) 5, 12

81. Zhang, Z., Liu, Y., Wang, X., Li, B., Hu, W.: Learn to match: Automatic matching network design for visual tracking. In: ICCV (2021) 11

82. Zhang, Z., Peng, H., Fu, J., Li, B., Hu, W.: Ocean: Object-aware anchor-free tracking. In: ECCV (2020) 11

83. Zheng, L., Tang, M., Chen, Y., Zhu, G., Wang, J., Lu, H.: Improving multiple object tracking with single object tracking. In: CVPR (2021) 11

84. Zhou, X., Koltun, V., Krähenbühl, P.: Tracking objects as points. In: ECCV (2020) 2, 3, 11

85. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. arXiv preprint arXiv:1904.07850 (2019) 3

86. Zhou, Z., Pei, W., Li, X., Wang, H., Zheng, F., He, Z.: Saliency-associated object tracking. In: ICCV (2021) 11

87. Zhu, J., Yang, H., Liu, N., Kim, M., Zhang, W., Yang, M.H.: Online multi-object tracking with dual matching attention networks. In: ECCV (2018) 4

88. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. In: ICLR (2020) 3, 6, 7, 14