Do Different Tracking Tasks Require Different Appearance Models?

Anonymous Author(s)
Affiliation
Address
email

Abstract

Tracking objects of interest in a video is one of the most popular and widely applicable problems in computer vision. However, with the years, a Cambrian explosion of use cases and benchmarks has fragmented the problem in a multitude of different experimental setups. As a consequence, the literature has fragmented too, and now the novel approaches proposed by the community are usually specialised to fit only one specific setup. To understand to what extent this separation is actually necessary, in this work we present **UniTrack**, a **Uni**fied **Track**ing solution to address *five* different tasks within the same framework. UniTrack consists of a single and task-agnostic appearance model, which can be learned in a supervised or self-supervised fashion, and multiple "heads" to address individual tasks and that do not require training. We show how most tracking tasks can be solved within this framework, and that the *same* appearance model can be used to obtain performance that is competitive against specialised methods, for all the five tasks considered. The framework also allows us to analyse appearance models obtained with the most recent self-supervised methods, thus significantly extending their evaluation and comparison to a larger variety of important problems.

1 Introduction

Unlike popular image-based computer vision tasks such as classification and object detection, which are (for the most part) unambiguous and clearly defined, the problem of *object tracking* has been treated under different setups and scenarios, each motivating a separate set of benchmarks and methods. For instance, for the Single Object Tracking (SOT) and Video Object Segmentation (VOS) communities [66, 25, 45], *tracking* means establishing the location of an arbitrary user-annotated target object throughout a video, where the location of the object is represented by a bounding-box in SOT and by a pixel-wise mask in VOS. Instead, in multiple object tracking settings (MOT [38], MOTS [53], and PoseTrack [2]), tracking means connecting sets of (given) detections across video frames to address the problem of identity associations and form trajectories. Despite these tasks only differ in the number of objects per frame to consider and observation format (bounding boxes, keypoints or masks), the best practices developed by the methods tackling them significantly vary.

Though on the one hand the proliferation of setups, benchmarks and methods is surely positive in that it allows specific use cases to be thoroughly studied, we argue it makes increasingly harder to effectively study the fundamental problems that all these tasks have in common, *i.e. what constitutes a good representation to track objects throughout a video? Could a single, universal representation address all different tracking setups?* Successes in (self-supervised) pre-trained models for natural language processing [51, 13] and vision [20, 8] suggest that we may not need different appearance models for different tracking tasks. In order to validate our speculation, in this paper we present a framework that allows to adopt the same appearance model to address *five* different tracking tasks

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

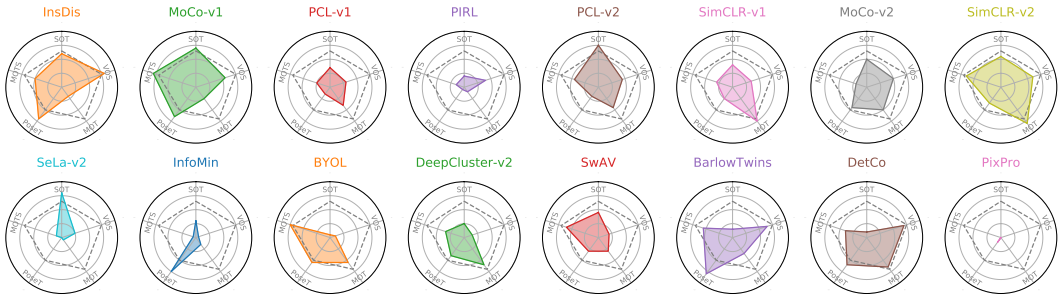


Figure 1: Radar charts of appearance model performance on five tracking tasks. We compute rankings in terms of AUC, $\mathcal{J}$-mean, IDF-1, IDF-1 and 1DF-1 for SOT, VOS, MOT, PoseTracking, and MOTS, respectively. A higher rank (better performance) corresponds to a vertex nearear the outer circle. Larger areas of the pentagons mean better overall performance. Results of ImageNet supervised model are shown with a gray dashed line.

(Figure 2). In our taxonomy (Figure 4), we consider existing tracking tasks as problems that have either *propagation* or *association* at their core. When the core problem is *propagation* (as in SOT and VOS), one has to localise a target object in the current frame given its location in the previous one. Instead, in association problems (MOT, MOTS, and PoseTrack), target states in both previous and current frames are given, and the goal is to determine the correspondence between the two sets of observations. We show how most tracking tasks currently considered by the literature can be simply expressed using these two components. For propagation tasks, we employ existing box and mask propagation algorithms [5, 57, 54]. For association tasks, we propose a novel reconstruction-based metric that leverages fine-grained correspondence to measure similarities between observations.

Finally, in our framework each individual task is mapped to a dedicated "head" that allows to represent the object(s) in the appropriate format to compare against prior art. Note that neither the propagation/association components nor the task-specific heads contain learnable parameters; only the base appearance model does. Importantly, we do not train appearance models on a single target task, nor apply multi-task learning on multiple target tasks, as training on specific datasets biases the learned representation towards a limited set of visual concepts (*e.g.* persons or vehicles), limiting its applicability to "open-world" settings. Instead, we employ universal and task-agnostic representations as our appearance model. These representations are supervised and self-supervised models popular in the community and that have already showed their effectiveness. This way, our work also serves the purpose of evaluating and comparing appearance models obtained from self-supervised learning approaches (see Figure 1) beyond the limited set of image-based problems typically seen in the literature. To summarise, the contributions of our work are as follows:

- We propose **UniTrack**, a framework that already supports five tracking tasks: SOT [66], VOS [45], MOT [38], MOTS [53], and PoseTrack [2], and that can be easily extended to new ones.
- We show how UniTrack can leverage many existing general-purpose appearance models to achieve a performance that is competitive with the state-of-the-art on all the five tasks considered.
- We propose a novel reconstruction-based similarity metric for association that preserves fine-grained visual features and supports multiple observation formats (box, mask and pose).
- We perform an extensive evaluation of self-supervised models, significantly extending their empirical analysis to video-based tasks.

2 The UniTrack Framework

2.1 Overview

Inspecting existing tracking tasks and benchmarks, we noticed that their differences can be categorised across four axes, illustrated in Figure 2 and detailed below.

1. Whether the requirement is to track a single (SOT [66, 25], VOS [45]) or multiple objects (MOT [45], MOTS [53], PoseTrack [2]).
2. Whether the targets are specified by a user in the first frame only (SOT, VOS), or instead are given in every frame, *e.g.* by a pre-trained detector (MOT, MOTS, PoseTrack).

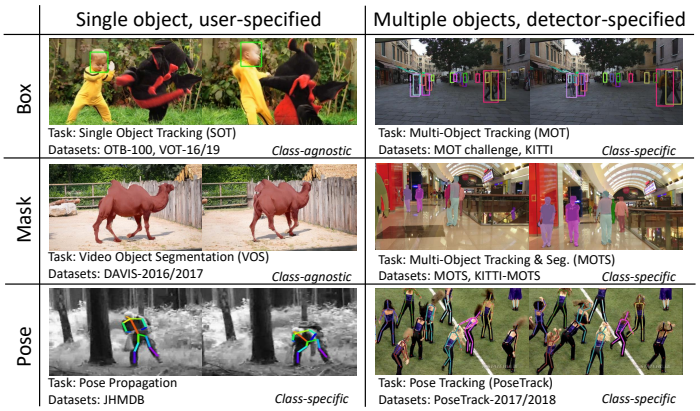


Figure 2: Existing tracking problems and their respective benchmarks differ from each other under several aspects: the assumption could be that there is a single or multiple objects to track; targets can be specified by the user in the first frame only, or assumed to be given at every frame (*e.g.* provided by a detector); the classes of the targets can be known (class-specific) or unknown (class-agnostic); the representation of the targets can be bounding boxes, pixel-wise masks, or pose annotations.

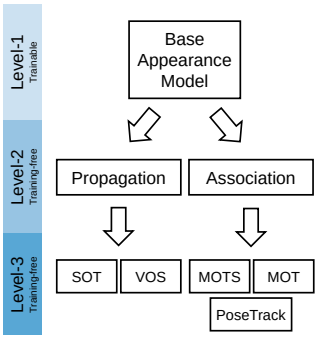


Figure 3: Overview of UniTrack. The framework can be divided in three levels. Level-1: a trainable base appearance model. Level-2: training-free algorithmic blocks. Level-3: training-free task-specific heads.

3. Whether the target objects are represented by bounding-boxes (SOT, MOT), pixel-wise masks (VOS, MOTS) or pose annotations (PoseTrack).
4. Whether the task is class-agnostic, *i.e.* target objects can be of *any* class (SOT, VOS); or if instead they are from a predefined set of classes (MOT, MOTS, PoseTrack)

Typically, in single-target tasks the target is specified by the user in the first frame, and it can be of any class. Instead, for multi-target tasks detections are generally considered as given for every frame, and the main challenge is to solve identity association for the several objects. Moreover, in multi-target tasks the set of classes to address is generally known (*e.g.* pedestrians or cars).

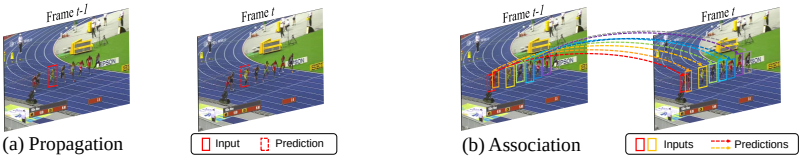
Figure 3 depicts a schematic overview of the proposed UniTrack framework, which can be understood as conceptually divided in three "levels". The first level is represented by the appearance model, responsible for extracting high-resolution feature maps (Section 2.2). The second level consists of the two fundamental algorithmic building blocks of *propagation* (Section 2.3) and *association* (Section 2.4). Finally, the last level comprises multiple task-specific algorithms that make direct use of the primitives of the second level. In this work, we illustrate how UniTrack can be used to obtain competitive performance on all of the five tracking tasks of level 3 from Figure 3. However, new tracking tasks can be easly integrated by using the framework and the primitives provided.

Importantly, note that the appearance model is the only component containing trainable parameters. The reason we opted for a *shared* and *non task-specific* representation is threefold. Firstly, the large amount of different setups motivated us to investigate whether it is necessary to fragment the problem of tracking in so many different specifications and have separately-trained models for each one; instead, we want to understand how far can a shared representation go. Next, we wanted to provide the community with multiple baselines that can be used to better assess newly proposed contributions. In other words, it would be important for a new task-specific method to significantly outperform ours, given its general-purpose design. Finally, we believe it is useful to have a versatile method that can be immediately used on new datasets and tasks without the need of retraining.

2.2 Base Appearance Model

The base appearance model ϕ takes as input a 2D image I and outputs a feature map $X = \phi(I) \in \mathbb{R}^{H \times W \times C}$. Since ideally an appearance model used for object propagation and association should be able to leverage fine-grained semantic correspondences between images, we choose a network with a small stride of $r = 8$, so that its output in feature space can have a relatively large resolution.

We refer to the vector (along the channels dimension) of a single point in the feature map as a point vector. Intuitively, we would expect a point vector $x_1^i \in \mathbb{R}^C$ from the feature map X_1 to highly resemble (in feature space) its "true match" point vector $x_2^{\hat{i}}$ in X_2, while being far apart from all



(a) Propagation (b) Association

Figure 4: Propagation *v.s.* Association. In the *propagation* problem, the goal is to estimate the target state at the current frame given the observation in the previous one. This is typically addressed for one object at the time. In the *association* problem, observations in both previous and current frames are given, and the goal is to determine correspondences between the two sets.

the "false match" point vectors x_2^j in X_2; *i.e.* we expect $s(x_1^i, x_2^{\hat{i}}) > s(x_1^i, x_2^j), \forall j \neq \hat{i}$, where $s(\cdot,\cdot)$ represents a similarity function.

In order to learn fine-grained correspondences, fully-supervised methods are only amenable for synthetic datasets (*e.g.* Flying Chairs for optical flow [14]). With real-world data, it is intractable to label pixel-level correspondences and train models in a fully-supervised fashion. To overcome this obstacle, in this paper we explore two sets of alternatives. The first is inspired by prior works that have pointed out how fine-grained correspondences emerge in middle-level features [35, 70]. Hence, a previously-trained model for classification or metric learning could serve us well as base appearance model. The second is inspired by recent progress in self-supervised learning approaches that specifically leverage pixel-wise pretext tasks [23, 54]. In this paper, we empirically explore which representations are best by measuring their performance on five different tracking problems.

2.3 Propagation

Problem definition. Figure 4a schematically illustrates the problem of *propagation*, which we use as a primitive to address SOT and VOS tasks. Considering the single-object case, given video frames $\{I_t\}_{t=1}^T$ and an initial ground truth observation z_1 as input, the goal is to predict object states $\{\hat{z}_t\}_{t=2}^T$ for each time-step t. In this work we consider three format of object states, namely bounding boxes, segmentation masks and pose skeletons.

Mask propagation. In order to propagate masks, we rely on an attention-based approach popularised by recent video self-supervised methods [23, 54, 31, 27]. Consider the feature maps of a pair of consecutive frames X_{t-1} and X_t, both $\in \mathbb{R}^{s \times C}$, and the label mask $z_{t-1} \in [0,1]^s$ of the previous frame [1], where $s = H \times W$ indicates its spatial resolution. We compute the matrix of transitions $K_{t-1}^t = [k_{i,j}]_{s \times s}$ as the affinity matrix between X_{t-1} and X_t. Each element $k_{i,j}$ is defined as

$$k_{i,j} = \text{Softmax}(X_{t-1} X_t^\top)_{ij} = \frac{\exp(\langle x_{t-1}^i, x_t^j \rangle / \tau)}{\sum_k^s \exp(\langle x_{t-1}^i, x_t^k \rangle / \tau)}, \tag{1}$$

where $\langle \cdot, \cdot \rangle$ indicates inner product, and τ is a temperature scalar. As in [23], we keep the top-K value in each row and set other values to zero. Then, the predicted mask for the current frame at time t is predicted simply by propagating the previous prediction: $z_t = K_{t-1}^t z_{t-1}$. Mask propagation proceeds in a recurrent fashion: the output mask of the current frame is used as input for the next one.

Pose propagation. In order to represent pose keypoints, we use the widely adopted Gaussian *belief maps* [62]. For a keypoint p, we obtain a belief map $z^p \in [0,1]^s$ by using a Gaussian with mean equal to the keypoint's location and variance proportional to the subject's body size. In order to propagate a pose, we can then individually propagate each belief map, again as $z_t^p = K_{t-1}^t z_{t-1}^p$.

Box propagation. The position of an object can also be simply expressed with a four-dimensional vector $z = (u, v, w, h)$, where (u, v) are the coordinates of the bounding-box center, and (w, h) are its width and height. Despite one could reuse the strategy adopted above by simply converting the bounding-box to a pixel-wise mask, we observed that using this strategy leads to inaccurate predictions. Instead, we use the approach of SiamFC [5], which consists in performing cross-correlation (XCorr) between the target template z_{t-1} and the frame X_t to find the new location of the target at frame t. Cross-correlation is performed at different scales, so that changes in target size can be detected and the bounding-box representation resized accordingly. We also provide a Deep Correlation Filter (DCF) [50, 57] based alternative, which also does not involve any training (see Appendix B.1).

[1] Note this corresponds to the ground-truth initialisation when $t = 1$, and to the latest prediction otherwise.

2.4 Association

Problem definition. Figure 4b schematically illustrates the *association* problem, which we use to address the tasks of MOT, MOTS and PoseTrack. In this case, observations for object states $\{\hat{\mathcal{Z}}_t\}_{t=1}^T$ are given for all the frames $\{I_t\}_{t=1}^T$, typically provided by a pre-trained detector. The goal here is to form trajectories by connecting observations across adjacent frames according to their identity.

Association algorithm. We adopt the association algorithm proposed in [61, 79] for MOT, MOTS and PoseTrack tasks, of which detailed description can be found in Appendix C. In summary, we compute an $N \times M$ distance matrix between N already-existing tracklets and M "new" detections from the last processed frame. We then use the Hungarian algorithm [26] to determine pairs of matches between tracklets and detections with the distance matrix as input. To obtain the matrix of distances used by the algorithm, we compute the linear combination of two terms accounting for *motion* and *appearance* cues. For the former, we compute a matrix indicating how likely a detection corresponds to the object state predicted by the Kalman Filter [24]. For the latter, the appearance component is computed using visual features from observations across frames: a given frame is first processed by the base appearance model (Section 2.2) to form a global feature map. Then, while box and mask observation features are directly obtained by cropping global feature maps, pose observations first need to be converted to masks (with a procedure described in Appendix C.2).

A key issue of this scenario is how to measure similarities between observation features. We find existing methods limited. First, feature maps are often compared by first performing global average pooling and then computing their cosine similarity [81, 48]. However, the process of averaging inherently discards local information. Second, simple metrics that do maintain fine-grained information, such as cosine similarity (of flattened feature maps) do not support varying spatial sizes like in the case of mask representations. To cope with the above limitations, we propose a reconstruction-based similarity metric that is able to deal with different observation formats while preserving fine-grained information.

Reconstruction Similarity Metric (RSM). Let $\{t_i\}_{i=1}^N$ denote the observation features of N existing tracklets, where each feature $t_i \in \mathbb{R}^{s_{t_i} \times C}$ and s_{t_i} indicates the spatial size of the object, *i.e.* the area of the box or the mask representing it. Similarly, $\{d_j\}_{j=1}^M$ denotes the observation features of M new detections. With the goal of measuring similarities to obtain an $N \times M$ affinity matrix to feed to the Hungarian algorithm, we propose a novel reconstruction-based similarity metric (RSM) between pairs (i, j), which is obtained as

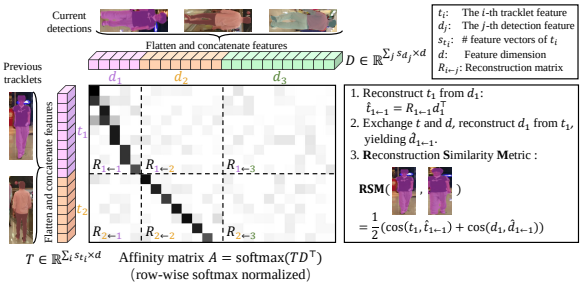


Figure 5: Illustration of the Reconstruction Similarity Metric.

$$\mathrm{RSM}(i, j) = \frac{1}{2}(\cos(t_i, \hat{t}_{i\leftarrow j}) + \cos(d_j, \hat{d}_{j\leftarrow i})), \tag{2}$$

where $\hat{t}_{i\leftarrow j}$ represents t_i reconstruction from d_j and $\hat{d}_{j\leftarrow i}$ represents d_j reconstruction from t_i. In multi-object tracking scenarios, observations are often incomplete due to frequent occlusions. As such, directly comparing features between incomplete and complete observations often fails, because the local content in observation features is misaligned. RSM remedies this issue by first aligning local features by reconstruction, and then calculating their similarity.

The reconstructed observation feature $\hat{t}_{i\leftarrow j}$ is a simple linear transformation of d_j, *i.e.* $\hat{t}_{i\leftarrow j} = R_{i\leftarrow j} d_j$, where $R_{i\leftarrow j} \in \mathbb{R}^{s_{t_i} \times s_{d_j}}$ is a transformation matrix needed to be calculated. To obtain $R_{i\leftarrow j}$, we first flatten and concatenate all tracklet observation features into a single feature matrix $T \in \mathbb{R}^{(\sum_i s_{t_i}) \times C}$. Similarly, we can obtain all the observation features of detections $D \in \mathbb{R}^{(\sum_j s_{d_j}) \times C}$. Finally, we extract $R_{i\leftarrow j}$ as the corresponding sub-matrix of A with respect to the (i, j) tracklet-detection pair, $R_{i\leftarrow j} = A\left[\sum_{i'=1}^{i-1} s_{i'} : \sum_{i'=1}^{i} s_{i'}, \sum_{j'=1}^{j-1} s_{j'} : \sum_{j'=1}^{j} s_{j'}\right]$ (see Figure 5), where A is an affinity matrix computed as $A = \mathrm{Softmax}(TD^\top)$.

RSM can be interpreted from an attention [51] perspective. The tracklet observation feature t_i being reconstructed can be seen as a set of *queries*, and the source detection feature d_j can be interpreted

both as *keys* and *values*. The goal is to compute the queries by linear combination of the keys. We first compute a *global* affinity matrix between t_i and *all* detections $d_{j'}$ for $j' = 1, ..., M$, and then extract the corresponding *local* sub-matrix as the attention matrix. Our formulation leads to a desired property: if t_i and d_j are dissimilar, the attention weights will likely approach zero, as the reconstructed $\hat{t}_i$ and the cosine similarity between t_i and d_j.

Measuring features similarity by reconstruction is popular in problems such as few-show learning [63, 77], self-supervised learning [34], and person re-identification [22]. However, reconstruction is typically framed as a ridge regression or optimal transport problem. RSM is more efficient than ridge regression, and it is computationally similar to calculate the Earth Moving Distance for the optimal transport problem. Appendix D shows a series of ablation studies illustrating the importance of the proposed RSM for the effectiveness of UniTrack on association-type tasks.

3 Experiments

Since UniTrack does not require training of specific downstream tasks, we are able to effectively experiment with many alternative *base appearance models* (see Figure 3). Surprisingly, we find out that off-the-shelf features obtained from an ImageNet pretrained architecture outperform most of the appearance models we benchmarked, on most tasks. For this reason, in Section 3.1 we use it as default base appearance model within UniTrack to compare against task-specific tracking methods. Then, in Section 3.2 we perform an extensive evaluation to benchmark a wide variety of modern self-supervised models, showing their strengths and weaknesses on all five tasks considered.

Implementation details. Unless specified, we adopt an ImageNet pre-trained ResNet-18 [21] as a default appearance model. To prevent excessive downsampling, we modify the spatial stride of `layer3` and `layer4` to 1, achieving a total stride of $r = 8$. We extract features from both `layer3` and `layer4`. We report results with `layer3` features when comparing against task-specific methods (Section 3.1), and with both `layer3` and `layer4` when evaluating multiple different representations (Section 3.2). Further implementation details are deferred to the Appendix B and C.

Datasets and evaluation metrics. For fair comparison with existing methods, we report results on standard benchmarks with conventional metrics for each task. Please refer to Appendix A for details.

3.1 Comparison with Task-specific Tracking Methods

Unsupervised methods. We observe that UniTrack performs favorably against unsupervised state-of-the-art methods in both propagation tasks we considered (Table 1a and 1b). For SOT, UniTrack with a DCF head [57] outperforms UDT [55] (a strong recent method) by 2.4 AUC percentage points, while it is surpassed by the LUDT+ [56] by 2.1 points. Considering that LUDT+ adopts an additional online template update mechanism [11] while ours does not, we believe the gap could be closed. In VOS, existing unsupervised methods are usually trained on video datasets [31, 23, 70]. Nonetheless, results show that even if our appearance model is pre-trained on still images, its tracking accuracy is still competitive. Note that, for association-type tasks, we are not aware of any existing unsupervised learning method, and thus in this case we limit the comparison to supervised methods.

Comparison with supervised methods. In general, UniTrack with a ResNet-18 appearance model already performs on par with several existing task-specific supervised methods, and in several tasks it even shows superior accuracy, especially for identity-related metrics. **(1)** For SOT, UniTrack with DCF head outperforms SiamFC [5] by 3.6 AUC points. This is a significant margin considering that SiamFC is trained with a large amount of crops from video datasets with annotated bounding boxes. **(2)** For VOS, UniTrack surpasses SiamMask [58] by 4.1 $\mathcal{J}$-mean points, despite this being trained on the joint set of three large-scale *video* datasets [33, 12, 71]. **(3)** For MOT, we employ the same detections used by the state-of-the-art tracker FairMOT [79]. The appearance embedding in FairMOT is trained with 270K bounding boxes of 8.7K labeled identities, from a MOT-specific dataset. In contrast, despite our appearance model not being trained with any MOT-specific data, our IDF1 score is quite competitive (71.8 *v.s.* 72.8 of FairMOT), and the ID switches are even reduced by 36.4%, from 1074 to 683. **(4)** For MOTS, we adopt identical observations as used in the COSTA [1] tracker, and observe a degradation in terms of ID switches (622 vs the 421 of the state of the art), and also a gap in IDF1 and sMOTA. **(5)** Finally, for pose tracking, we employ the identical still-image pose estimator used by LightTrack [40]. Compared with LightTrack, the MOTA of UniTrack degrades of

Methods	AUC↑
Supervised:	
SiamFC [5]	58.2
SiamRPN [29]	63.7
SiamRPN++ [28]	**69.6**
Unsupervised:	
UDT [55]	59.4
UDT+ [55]	63.2
LUDT [56]	60.2
LUDT+ [56]	**63.9**
UniTrack_XCorr	55.5
UniTrack_DCF	61.8

(a) SOT@OTB-2015 [66].

Methods	$\mathcal{J}$-mean↑
Supervised:	
SiamMask [58]	54.3
FEELVOS [52]	63.7
STM [42]	**79.2**
Unsupervised:	
Colorization [54]	34.6
TimeCylce [60]	40.1
UVC [31]	56.7
CRW [23]	64.8
VFS [70]	**66.5**
UniTrack	58.4

(b) VOS@DAVIS-2017 [45].

Methods	IDF1↑	IDs↓	MOTA↑
POI [75]	65.1	805	66.1
DeepSort-2 [64]	62.2	781	61.4
JDE [61]	55.8	1544	64.4
CTracker [44]	57.2	1897	67.6
TubeTK [43]	62.2	1236	66.9
MAT [19]	63.8	928	73.5
TraDes [65]	64.7	1144	70.1
CSTrack [32]	71.8	1071	70.7
FairMOT† [79]	**72.8**	1074	**74.9**
UniTrack†	71.8	**683**	74.7

(c) MOT@MOT-16 [38] *test* split, private detection.

Methods	IDF1↑	IDs↓	sMOTA↑
TrackRCNN [53]	42.4	567	40.6
SORTS [64]	57.3	577	55.0
PointTrack [72]	42.9	868	62.3
GMPHD [47]	65.6	566	69.0
COSTA† [1]	**70.3**	**421**	**70.2**
UniTrack†	67.2	622	68.9

(d) MOTS@MOTS [53] *test* split.

Methods	IDF1↑	IDs↓	MOTA↑
MDPN [18]	-	-	50.6
OpenSVAI [41]	-	-	62.4
Miracle [74]	-	-	64.0
KeyTrack [46]	-	-	**66.6**
LightTrack† [40]	52.2	**3024**	64.8
UniTrack†	**73.2**	6760	63.5

(e) PoseTrack@PoseTrack2018 [2] *val* split.

Table 1: Comparison with task-tailored unsupervised and supervised methods on five typical tracking tasks. † indicates methods using identical observations.

1.3 points because of an increased amount of ID switches. However, the IDF-1 score is improved by a significant margin (+21.0 points), showing that UniTrack preserves identity much more accurately in long-term tracking.

3.2 UniTrack as a Representation Evaluation Platform

The evaluation of representations learned via self-supervised learning (SSL) often involves additional training [15, 20, 8], for instance by the use of *linear probes* [78]. In contrast, UniTrack can be used as a representations evaluation platform that **(1)** does not require any further training and **(2)** tests on a battery of important video tasks, which have generally been neglected by SSL works in favour of evaluation on standard tasks such as classification. In this section, we evaluate three types of SSL representations: **(a)** Middle-level features of image representations trained with translation-invariant pretext tasks, *e.g.* MoCo [20] and BYOL [17]; **(b)** Pixel-level representations trained for non-translation-invariant pretext tasks using still images (such as DetCo [68] and PixPro [69]) and **(c)** videos (such as UVC [31] and CRW [23])/ For all methods considered, we use the pre-trained weights of a ResNet-50 or ResNet-18 architecture provided by the authors. Results are shown in Table 2 and 3. For each of the five tasks and multiple metrics considered, we report results obtained from features both at `layer3` and `layer4` (separated by a '/' in the table) of the ResNet architecture. Darker gradients of red represent progressively better performance (relative to that column). For each performance measure, the best value is underlined. Note that, for this analysis, for association tasks the motion cues are discarded to better highlight distinctions between different representations and avoid potential confounding factors. Figure 1 provides a high-level summary of the results by focusing on the ranking obtained by different SSL methods on the five tracking tasks considered (each represented by a vertex in the radar-style plot). Several observations can be made:

(1) *There is no significant correlation between "linear probe accuracy" on ImageNet and overall tracking performance.* The linear-probe approach [78] has became fairly standard as *the* way to compare SSL representations. In Figure 6, we plot tracking performance on five tasks (y-axes) against ImageNet top-1 accuracy of 16 different models (x-axes), and report Pearson and Spearman (rank) correlation coefficients. We observe that the correlation between ImageNet accuracy and tracking performance is small, *i.e.* the Pearson's r ranges from -0.38 to $+0.20$, and Spearman's ρ ranges from -0.36 to $+0.26$. For most tasks, there is almost no correlation, while for VOS the two measures are mildly *inversely* correlated. The result suggests that evaluating SSL models on five extra tasks with UniTrack could constitute a useful complement to ImageNet linear probe evaluation, and encourage the SSL community to pursue the design of even more *general purpose* representations.

(2) *A vanilla ImageNet-trained representation is surprisingly effective.* On most tasks, it reports a performance that is on par with the best representation for that task. This is particularly evident from

Representation	SOT [66]		VOS [45]	MOT [38]		MOTS [53]		PoseTrack [2]	
	AUC_{XCorr} ↑	AUC_{DCF} ↑	$\mathcal{J}$-mean↑	IDF1↑	HOTA↑	IDF1↑	HOTA↑	IDF1↑	IDs↓
Rand. Init.	**10.3** / 9.0	**28.0** / 20.0	29.3 / **33.9**	8.4 / **8.9**	8.4 / **8.5**	20.8 / **23.1**	25.9 / **28.7**	**40.2** / 38.5	**88792** / 90963
Supervised	**58.6** / 49.5	**62.0** / 53.9	**62.3** / 57.9	**75.6** / 73.2	**63.3** / 61.8	68.4 / **69.4**	70.2 / **71.0**	**73.7** / 73.3	**6969** / 7103
InsDis [67]	**47.6** / 47.3	**61.8** / 51.1	**62.6** / 60.1	66.7 / **73.9**	57.9 / **61.9**	**68.4** / 68.0	69.6 / **70.3**	72.4 / **73.9**	7106 / **7015**
MoCoV1 [20]	**50.9** / 47.9	**62.2** / 53.7	**61.5** / 57.9	69.2 / **74.1**	59.4 / **61.9**	**70.6** / 69.3	**71.6** / 70.9	72.8 / **73.9**	**6872** / 7092
PCLV1 [30]	**56.8** / 31.5	**61.3** / 35.0	**60.4** / 38.8	**74.8** / 68.8	**62.8** / 59.1	**67.6** / 65.2	**69.7** / 67.3	**73.3** / 71.1	**6855** / 10694
PIRL [39]	43.8 / **51.0**	**61.2** / 53.4	**60.8** / 57.7	62.0 / **73.4**	54.6 / **61.9**	66.0 / **67.4**	66.7 / **69.9**	72.1 / **73.0**	7235 / **7173**
PCLV2 [30]	**54.9** / 50.3	**62.5** / 51.6	**61.2** / 52.5	**74.9** / 72.9	**62.7** / 61.8	**68.3** / 66.6	**70.5** / 69.0	**73.5** / 73.4	**6859** / 8489
SimCLRV1 [8]	47.3 / **51.9**	**61.3** / 50.7	**60.5** / 56.5	66.9 / **75.6**	57.7 / **63.2**	65.8 / **67.6**	67.7 / **69.5**	72.3 / **73.5**	**7084** / 7367
MoCoV2 [10]	**53.7** / 47.2	**61.5** / 53.3	**61.2** / 54.0	72.0 / **74.9**	61.2 / **62.8**	**67.5** / 67.3	**69.6** / **69.6**	73.0 / **73.7**	**6932** / 7702
SimCLRV2 [9]	50.0 / **54.7**	**61.7** / 56.8	**61.6** / 58.4	67.6 / **75.7**	58.1 / **63.3**	**69.1** / 67.4	**70.4** / 69.4	72.5 / **73.6**	**7228** / 7856
SeLaV2 [3]	**51.0** / 9.6	**63.1** / 14.2	**60.2** / 40.2	68.8 / **68.9**	59.0 / **59.3**	**66.8** / 66.1	**68.7** / 68.5	**72.9** / 72.3	**6983**/ 7815
Infomin [49]	**48.5** / 46.8	**61.2** / 51.9	**58.4** / 51.1	66.7 / **73.4**	57.6 / **61.9**	**66.7** / 66.3	68.5 / **68.8**	72.5 / **74.0**	**7066** / 7901
BarLowTwins [7	44.5 / **55.5**	**60.5** / 60.1	**61.7** / 57.8	63.7 / **74.5**	55.4 / **62.4**	**68.7** / 67.4	69.5 / **69.8**	72.3 / **74.3**	**7131** / 7456
BYOL [17]	48.3 / **55.5**	**58.9** / 56.8	**58.8** / 54.3	65.3 / **74.9**	56.8 / **62.9**	**70.1** / 66.8	**70.8** / 69.3	72.4 / **73.8**	**7213** / 8032
DeepClusterV2 [6]	51.5 / **52.9**	**61.2** / **61.2**	**59.3** / 53.4	66.9 / **75.1**	57.8 / **63.5**	**67.7** / 67.4	69.4 / **69.8**	72.7 / **73.7**	**7018** / 7283
SwAV [7]	49.2 / **52.4**	**61.5** / 59.4	**59.4** / 57.0	65.6 / **74.4**	56.9 / **62.3**	**68.8** / 67.0	**69.9** /69.5	72.7 / **73.6**	**7025** / 7377
PixPro [69]	40.5 / **49.2**	**57.4** / 49.3	**56.4** / 52.2	61.7 / **67.7**	54.3 / **58.6**	64.2 / **66.2**	65.1 / **67.6**	72.4 / **73.1**	7163 / **6953**
DetCo [68]	**55.0** / 47.1	**59.0** / 53.2	**62.3** / 56.1	**75.3** / 72.9	**62.8** / 61.6	**67.8** / 66.8	**70.0** / 69.4	**73.9** / 73.3	**7357** / 8009
TimeCycle [60]	**43.8** / 24.2	**57.5** / 48.7	**51.8** / 48.9	**68.7** / 28.2	**59.3** / 25.5	**69.9** / 47.1	**71.3** / 49.3	**72.0** / 62.3	**7837** / 27884

Table 2: Tracking performance of pre-trained *image-based* SSL models. All methods employ ResNet-50.

Representation	SOT [66]		VOS [45]	MOT [38]		MOTS [53]		PoseTrack [2]	
	AUC_{XCorr} ↑	AUC_{DCF} ↑	$\mathcal{J}$-mean↑	IDF1↑	HOTA↑	IDF1↑	HOTA↑	IDF1↑	IDs↓
Rand. Init.	16.0 / **18.2**	**36.1** / 32.1	33.0 / **36.7**	**18.4** / 14.6	**20.2** / 12.9	**34.5** / 33.1	**39.9** / 37.6	**52.8** / 50.5	**65317** / 66230
Supervised	**55.0** / 46.2	**61.8** / 52.6	**58.4** / 46.7	**74.8** / 74.5	**62.7** / 62.1	67.6 / **68.6**	69.8 / **70.5**	72.7 / **73.2**	**6808** / 7024
Color. [54]+mem.	41.6 / **43.4**	56.7 / **58.7**	53.6 / **59.7**	**64.9** / 62.8	**56.8** / 55.5	**68.8** / 66.1	**69.4** / 66.3	72.4 / **72.6**	6850 / **6778**
UVC [31]	**46.0** / 38.7	58.1 / **59.9**	**56.5** / 53.9	**66.9** / 64.5	**57.7** / 54.1	**69.9** / 68.7	**69.6** / 69.4	72.6 / **72.8**	**6843** / 6972
CRW [23]	46.3 / **49.1**	**58.9** / 54.9	**63.2** / 60.7	67.8 / **73.0**	58.4 / **61.7**	69.0 / **71.3**	69.2 / **71.9**	72.7 / **73.0**	6799 / **6761**

Table 3: Tracking performance of pre-trained *video-based* SSL models. All methods employ ResNet-18. In the above two tables, we report results with [`layer3` / `layer4`] features in each cell, and the best performance between the two is **bolded**. We use the bolded value as the value to rank the models in each column, and visualize better performance with darker cell colors. Best results in each column are underlined.

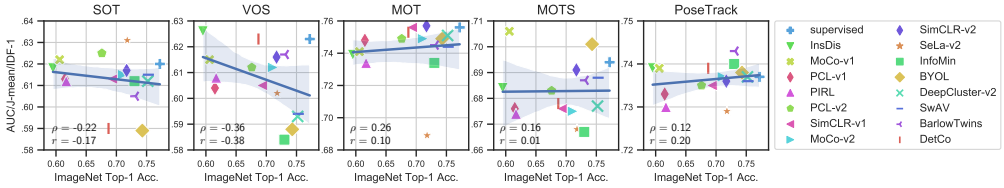


Figure 6: Tracking performance is poorly correlated with ImageNet accuracy. On the x-axes we plot ImageNet linear probe top-1 accuracy and on the y-axes the tracking performance on five tracking datasets. Correlation coefficients (Spearman's ρ and Pearson's r) are shown in the left bottom of each plot.

Figure 1, where its performance is outlined as a gray dashed line. This result suggests that results obtained with vanilla ImageNet features should be reported when investigating new tracking methods. An important contribution of our work is to make this possible with the UniTrack framework.

(3) *No SSL representation dominates all tasks.* Recently, it has been shown how SSL-trained representations can match or surpass their supervised counterparts on ImageNet linear classification (*e.g.* [17]) and many downstream tasks [15, 68]. Similarly, within UniTrack the best SSL representation matches or beats supervised models on every single task. However, no individual SSL representation is able to beat the vanilla ImageNet-trained representation on every task. Among SSL methods, two emerge as clear "winners" from this comparison. One is SimCLR-v2 [9], which performs almost as well as the vanilla ImageNet-trained representation on three tasks and outperforms it on MOT. The other is MoCo [20], which despite being one of the worst methods on MOT, is able to outperform the vanilla ImageNet-trained representation on SOT, MOTS and PoseTrack. Curiously, MoCo-v2 [10] has an overall worse performance than its predecessor.

(5) *Pixel-level SSL representations do not have a consistent advantage in pixel-level tasks.* In Table 3 and at the bottom of Table 2 we compare recent SSL representations trained with pixel-level proxy tasks: PixPro [69], DetCo [68], TimeCycle [60], Colorization [54], UVC [31] and Contrastive Random Walk (CRW) [23]. Considering that pixel-level models leverage more fine-grained information during training, one may expect them to outperform image-based models in the tracking tasks where this is important. It is not straightforward to compare pixel-level SSL models with image-level ones, as the two types employ different default backbone networks. However, note how good image-based models (MoCo-v1, SimCLR-v2) are on par with their supervised counterpart in all

tasks, while good pixel-level models (DetCo, CRW) still have gaps to their supervised counterpart in tasks like SOT and MOT. Moreover, from Table 2, one can notice how the last three rows, despite representing methods leveraging pixel-level information during training, are actually outperformed by image-level representations on the pixel-level tasks of VOS, MOTS and PoseTrack.

(6) *SSL representations trained from video do not have a consistent advantage overall.* Beside being trained with pixel-level proxy tasks, TimeCycle [60], Colorization [54], UVC [31] and CRW [23] are also trained on large video datasets, which should intuitively provide an advantage for video-based tasks like the five considered in this paper. Instead, with the exception of CRW [23] (which is the performer for VOS and MOTS), they are generally outperformed by ImageNet-trained features.

4 Related Work

Sharing appearance model across multiple tracking tasks has not yet being extensively studied in literature, especially not in the context of SSL representations. Some existing methods do share a common backbone architecture across tasks. For instance, STEm-Seg [4] addresses VIS [73] and MOTS; while TraDeS [65] addresses 2D/3D MOT, MOTS and VIS. However, both methods need to be trained separately and on different datasets for every task. Conversely, we reuse the same representation across five tasks and we do not require any training. A promising direction for future work would be to use UniTrack to train a shared representation in a multi-task fashion. Only a few relevant works do adopt a multi-task approach [58, 80, 37], and they usually consider SOT and VOS tasks only. In general, despite the multi-task direction being surely interesting, it requires the availability of large-scale datasets with annotations in multiple formats, and costly training. These are two of the main reasons for which we believe that having a framework that allows to achieve competitive performance on multiple tasks with previously-trained models is a worthwhile endeavour.

Self-supervised model evaluation. Given the difference between the pretext tasks used to train self-supervised models and the downstream tasks used to evaluate them, the comparison between self-supervised approaches has always been a delicate matter. Existing evaluation strategies typically require additional training once a general-purpose representation has been obtained. One strategy keeps the representation fixed, and then trains additional task-specific heads with very limited capacity (*e.g.* a linear classifier [16, 8, 20] or a regression head for object detection [16]). A second strategy, instead, leverages SSL to obtain particularly effective initializations, and then proceeds to fine-tune such initialized models on the downstream task of interest. A wider range of tasks can be tested using this setup, such as semantic segmentation [15, 16] and surface normal estimation [16, 59]. In contrast, UniTrack provides a simpler way to evaluate SSL models, one that does not require additional training or fine-tuning. Also, this work is the first to extend SSL evaluation to a set of diverse video tasks. We believe this contribution will allow the study of self-supervised method with a broader scope of applicability. Our work is also related to a line of self-supervised learning methods [23, 54, 31, 27] that learn their representations in a task-agnostic fashion, and then test it on *propagation* tasks (SOT and VOS). The design of UniTrack is inspired by their task-agnostic philosophy, while significantly extending their scope to a new set of tasks.

5 Conclusion

Do different tracking tasks require different appearance models? In order to address this question, the proposed UniTrack framework has been instrumental, as it has allowed to easily experiment with alternative representations on a wide variety of downstream problems. From our study, the answer is negative: it turns out that a *single shared* appearance model can perform competitively against dedicated methods, without any training. Our findings point towards the existence of several strong baselines which should be carefully considered when designing task-specific approaches. Moreover, our work also allows to significantly extend the evaluation of self-supervised models to five new tasks that, as we demonstrated in our experiments, have little correlation with the standard linear-probe approach, and thus high informative value. We believe this will encourage the community to develop self-supervised representations that are of "general purpose" in a broader sense.

Broader impact. Upon reflection, we believe progress in tracking applications and self-supervised learning is beneficial for society, as it can significantly positively impact progress in the field of autonomous vehicles, for instance. We also recognise that the same technologies might constitute a threat if deployed for surveillance (*e.g.* identity recognition) by entities hostile to civil liberties.

References

[1] Costa_st tracker. https://motchallenge.net/method/MOTS=87&chl=17.

[2] Mykhaylo Andriluka, Umar Iqbal, Eldar Insafutdinov, Leonid Pishchulin, Anton Milan, Juergen Gall, and Bernt Schiele. Posetrack: A benchmark for human pose estimation and tracking. In *CVPR*, 2018.

[3] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *ICLR*, 2020.

[4] Ali Athar, Sabarinath Mahadevan, Aljoša Ošep, Laura Leal-Taixé, and Bastian Leibe. Stem-seg: Spatio-temporal embeddings for instance segmentation in videos. In *ECCV*, 2020.

[5] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *ECCV workshops*, 2016.

[6] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018.

[7] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2018.

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. 2020.

[9] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv:2006.10029*, 2020.

[10] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv:2003.04297*, 2020.

[11] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Eco: Efficient convolution operators for tracking. In *CVPR*, 2017.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 2018.

[14] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015.

[15] Linus Ericsson, Henry Gouk, and Timothy M Hospedales. How well do self-supervised models transfer? In *CVPR*, 2021.

[16] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. *arXiv:1905.01235*, 2019.

[17] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv:2006.07733*, 2020.

[18] Hengkai Guo, Tang Tang, Guozhong Luo, Riwei Chen, Yongchen Lu, and Linfu Wen. Multi-domain pose network for multi-person pose estimation and tracking. In *ECCV, workshop*, 2018.

[19] Shoudong Han, Piao Huang, Hongwei Wang, En Yu, Donghaisheng Liu, Xiaofeng Pan, and Jun Zhao. Mat: Motion-aware multi-object tracking. *arXiv:2009.04794*, 2020.

[20] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[22] Lingxiao He, Jian Liang, Haiqing Li, and Zhenan Sun. Deep spatial feature reconstruction for partial person re-identification: Alignment-free approach. In *CVPR*, 2018.

[23] Allan Jabri, Andrew Owens, and Alexei A Efros. Space-time correspondence as a contrastive random walk. In *NeurIPS*, 2020.

[24] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.

[25] Matej Kristan, Jiri Matas, Aleš Leonardis, Tomáš Vojíř, Roman Pflugfelder, Gustavo Fernandez, Georg Nebehay, Fatih Porikli, and Luka Čehovin. A novel performance evaluation methodology for single-target trackers. *IEEE TPAMI*, 2016.

[26] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[27] Zihang Lai, Erika Lu, and Weidi Xie. Mast: A memory-augmented self-supervised tracker. In *CVPR*, 2020.

[28] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *CVPR*, 2019.

[29] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *CVPR*, 2018.

[30] Junnan Li, Pan Zhou, Caiming Xiong, Richard Socher, and Steven CH Hoi. Prototypical contrastive learning of unsupervised representations. In *ICLR*, 2021.

[31] Xueting Li, Sifei Liu, Shalini De Mello, Xiaolong Wang, Jan Kautz, and Ming-Hsuan Yang. Joint-task self-supervised learning for temporal correspondence. In *NeurIPS*, 2019.

[32] Chao Liang, Zhipeng Zhang, Yi Lu, Xue Zhou, Bing Li, Xiyong Ye, and Jianxiao Zou. Rethinking the competition between detection and reid in multi-object tracking. *arXiv:2010.12138*, 2020.

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

[34] Songtao Liu, Zeming Li, and Jian Sun. Self-emd: Self-supervised object detection without imagenet. *arXiv:2011.13677*, 2020.

[35] Jonathan Long, Ning Zhang, and Trevor Darrell. Do convnets learn correspondence? In *NeurIPS*, 2014.

[36] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *IJCV*, 2020.

[37] Alan Lukezic, Jiri Matas, and Matej Kristan. D3s-a discriminative single shot segmentation tracker. In *CVPR*, 2020.

[38] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv:1603.00831*, 2016.

[39] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *CVPR*, 2020.

[40] Guanghan Ning and Heng Huang. Lighttrack: A generic framework for online top-down human pose tracking. In *CVPR, workshop*, 2020.

[41] Guanghan Ning, Ping Liu, Xiaochuan Fan, and Chi Zhang. A top-down approach to articulated human pose estimation and tracking. In *ECCV, workshop*, 2018.

[42] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, 2019.

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

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

[45] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016.

[46] Michael Snower, Asim Kadav, Farley Lai, and Hans Peter Graf. 15 keypoints is all you need. In *CVPR*, 2020.

[47] Young-min Song and Moongu Jeon. Online multi-object tracking and segmentation with gmphd filter and simple affinity fusion. *arXiv:2009.00100*, 2020.

[48] Yifan Sun, Qin Xu, Yali Li, Chi Zhang, Yikang Li, Shengjin Wang, and Jian Sun. Perceive where to focus: Learning visibility-aware part-level features for partial person re-identification. In *CVPR*, 2019.

[49] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning. In *NeurIPS*, 2020.

[50] Jack Valmadre, Luca Bertinetto, Joao Henriques, Andrea Vedaldi, and Philip HS Torr. End-to-end representation learning for correlation filter based tracking. In *CVPR*, 2017.

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

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

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

[54] Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. Tracking emerges by colorizing videos. In *ECCV*, 2018.

[55] Ning Wang, Yibing Song, Chao Ma, Wengang Zhou, Wei Liu, and Houqiang Li. Unsupervised deep tracking. In *CVPR*, 2019.

[56] Ning Wang, Wengang Zhou, Yibing Song, Chao Ma, Wei Liu, and Houqiang Li. Unsupervised deep representation learning for real-time tracking. *IJCV*, 2020.

[57] Qiang Wang, Jin Gao, Junliang Xing, Mengdan Zhang, and Weiming Hu. Dcfnet: Discriminant correlation filters network for visual tracking. *arXiv:1704.04057*, 2017.

[58] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *CVPR*, 2019.

[59] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *ICCV*, 2015.

[60] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *CVPR*, 2019.

[61] Zhongdao Wang, Liang Zheng, Yixuan Liu, and Shengjin Wang. Towards real-time multi-object tracking. In *ECCV*, 2020.

[62] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, 2016.

[63] Davis Wertheimer, Luming Tang, and Bharath Hariharan. Fine-grained few-shot classification with feature map reconstruction networks. *CVPR*, 2021.

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

[65] Jialian Wu, Jiale Cao, Liangchen Song, Yu Wang, Ming Yang, and Junsong Yuan. Track to detect and segment: An online multi-object tracker. In *CVPR*, 2021.

[66] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *CVPR*, 2013.

[67] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018.

[68] Enze Xie, Jian Ding, Wenhai Wang, Xiaohang Zhan, Hang Xu, Zhenguo Li, and Ping Luo. Detco: Unsupervised contrastive learning for object detection. *arXiv:2102.04803*, 2021.

[69] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *CVPR*, 2021.

[70] Jiarui Xu and Xiaolong Wang. Rethinking self-supervised correspondence learning: A video frame-level similarity perspective. *arXiv:2103.17263*, 2021.

[71] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *ECCV*, 2018.

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

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

[74] Dongdong Yu, Kai Su, Jia Sun, and Changhu Wang. Multi-person pose estimation for pose tracking with enhanced cascaded pyramid network. In *ECCV, workshop*, 2018.

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

[76] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv:2103.03230*, 2021.

[77] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *CVPR*, 2020.

[78] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *CVPR*, 2017.

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

[80] Zhipeng Zhang, Bing Li, Weiming Hu, and Houweng Peng. Towards accurate pixel-wise object tracking by attention retrieval. *arXiv:2008.02745*, 2020.

[81] Kuan Zhu, Haiyun Guo, Zhiwei Liu, Ming Tang, and Jinqiao Wang. Identity-guided human semantic parsing for person re-identification. In *ECCV*, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Please see Appendix E in the supplementary material.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Please see Section 5
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] The code will be released upon acceptance.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Please see Appendix A for dataset splits, Appendix B and C for hyperparameters.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Please see Appendix A.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] Please see Appendix A.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] Please see Appendix A.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]