Learning to Track Instances without Video Annotations

Yang Fu[1*], Sifei Liu[2], Umar Iqbal[2], Shalini De Mello[2]
Humphrey Shi[1,3†], Jan Kautz[2]
[1]University of Illinois at Urbana-Champaign, [2]NVIDIA, [3]University of Oregon

Abstract

Tracking segmentation masks of multiple instances has been intensively studied, but still faces two fundamental challenges: 1) the requirement of large-scale, frame-wise annotation, and 2) the complexity of two-stage approaches. To resolve these challenges, we introduce a novel semi-supervised framework by learning instance tracking networks with only a labeled image dataset and unlabeled *video sequences. With an instance contrastive objective, we learn an embedding to discriminate each instance from the others. We show that even when only trained with images, the learned feature representation is robust to instance appearance variations, and is thus able to track objects steadily across frames. We further enhance the tracking capability of the embedding by learning correspondence from unlabeled videos in a self-supervised manner. In addition, we integrate this module into single-stage instance segmentation and pose estimation frameworks, which significantly reduce the computational complexity of tracking compared to two-stage networks. We conduct experiments on the YouTube-VIS and PoseTrack datasets. Without any video annotation efforts, our proposed method can achieve comparable or even better performance than most fully-supervised methods*[1].

1. Introduction

In recent years, the vision community has rapidly improved the performance of instance segmentation at both the image and video levels as a core technique in autonomous driving. The pipeline for segmenting instances from videos commonly includes: (i) segmentation on individual frame; and (ii) linking of each instance across frames for an entire video sequence. Most existing approaches [5, 8, 23, 43] employ fully-supervised learning that relies on dense annotations of instance segmentation masks and instance associations across video frames (see Fig. 1 top). Since annotation of videos, especially in a per-frame manner requires excessive labor, the fully-supervised learning setting, however, becomes the major bottleneck for frame-wise video processing.

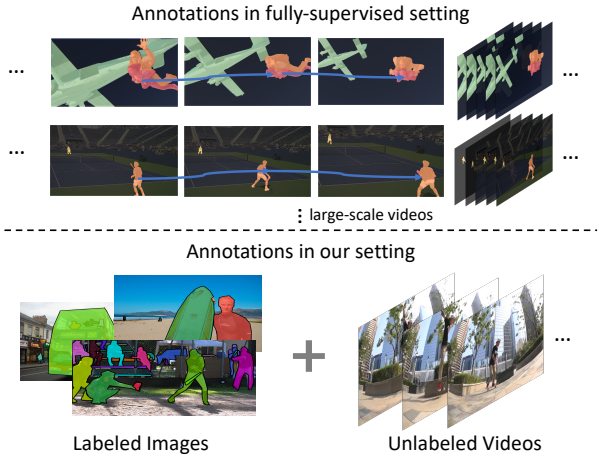

Figure 1. The annotations required for our proposed approach vs. those for fully supervised approaches.

To reduce the dependence on labels, self-supervised tracking approaches have been developed to learn pixel-level video correspondences from large-scale unlabeled videos [17, 20, 39]. The learned correspondences can be used to track any fine-grained attributes, *e.g.*, segmentation masks, keypoints and textures, on a per-pixel basis. However, such self-supervised approaches aim to learn semantically-independent representations, *i.e.*, they do not discriminate between object instances. Such approaches can be used for tracking only when ground truth attributes are annotated at keyframes, *e.g.*, the 1st frame of any sequence [28]; or when additional pre-trained instance segmentation models are provided.

In this paper, we consider a novel semi-supervised setting: we learn to track instances only with a labeled image dataset, and optionally, unlabeled video sequences. In other words, in addition to learning image-level instance segmentation, we also learn to associate instances across frames in a self-supervised manner. Our setting strikes a balance

---

∗ This work was done while Yang Fu was a research intern at NVIDIA
† corresponding author
[1]Project page: https://oasisyang.github.io/projects/semi-track/index.html

between the fully-supervised and the self-supervised ones. With regards to it applications, our model can be seamlessly adapted and utilized for tracking objects on newly captured videos, *e.g*., traffic scene sequences during autonomous driving, without requiring any offline processing.

A typical way to learn tracking is to model instance association as a multi-class classification problem [43]. Since we do not have the ground truth association labels, we instead learn a feature map that should be: (i) discriminative of different instances, and (ii) robust to appearance variation caused by motion of instances in videos. Once learned, any object instance can be tracked by utilizing its feature embedding to search for the most similar one in the next frame. To learn it with only labeled images, we introduce an Instance Contrastive (IC) objective defined densely on the embedding map. This objective encourages the pixel-level feature embedding to be consistent when being sampled from the same instance, while being less consistent for different ones. In addition, we optimize a Maximum Entropy (ME) regularization to enforce that each instance, on being matched to others, exhibits a uniform distribution. With this constraint, when a new object enters a sequence, the model can easily detect it by comparing it with all existing instances , and thus assign it a new instance label.

In addition to using labeled images, we also discover when leveraging unlabeled videos, tracking performance can be further improved via self-supervised learning. In this work, we choose to learn self-supervised video correspondences. Specifically, we adopt a cycle-consistency loss by maximizing the likelihood of pixels returning to their original location on being propagated forward and backward along a stack of frames [17]. Since the feature embedding is utilized to construct the cross-frame affinity for propagation, it can be implicitly enhanced by enforcing this objective. Intuitively, video correspondence learning improves tracking performance by potentially encouraging the network to "see" more instance appearance variations in time.

To further mitigate the data distribution shifts between labeled images, unlabeled videos, and testing videos, we introduce a self-supervised test-time adaptation strategy. Inspired by [33], we enhance the model's tracking capability by keeping the self-supervised objective at the inference stage, and adapting it to any particular input sequence.

Instead of learning an independent network that separately produces the feature embedding for tracking, we integrate it as a head in to a bottom-up instance segmentation framework, *e.g*., SOLO [40]. With labeled images, we jointly train the instance segmentation and the feature embedding parts of the network, enriching the original network with the new function of tracking. We note that in addition to introducing a semi-supervised setting, we are also propose a bottom-up framework for tracking masks of multiple instances. Finally, we also show that similar approaches can be generalized to the task of multiple human pose tracking, when building on top of a bottom-up human pose estimation network [42]. In summary, we conclude our contribution as the following:

- A novel semi-supervised setting that can largely reduce the effort of labelling large-scale video datasets.
- An Instance Contrastive loss equipped with Maximum Entropy regularization to learn a feature embedding capable of tracking with only labeled images.
- A self-supervised video correspondence learning method that further improves tracking performance by leveraging unlabelled videos.
- Extensive experiments demonstrate that the proposed method performs on par if not better than most state-of-the-arts approaches, for both the video instance segmentation and pose tracking tasks.

2. Related Work

Video Instance Segmentation is the joint task of detection, segmentation and tracking of object instances in videos. MaskTrack-RCNN [43] is the first attempt to address the video instance segmentation problem. It proposes a large-scale video dataset named YouTube-VIS for benchmarking video instance segmentation algorithms. MaskTrack RCNN extends Mask RCNN [15] with an additional tracking branch and achieves object association by object embedding and other cues, *i.e*., position and category. In addition, several methods from the Large-Scale Video Object Segmentation Challenge [1] achieve impressive results with large quantities of external data and complex algorithmic pipelines [8, 23, 38, 10]. However, all these mentioned approaches heavily depend on video annotations, and to the best of our knowledge, our method is the first attempt at video instance segmentation without any video annotations.

Contrastive Learning has recently received interest due to its success in self-supervised representation learning in the computer vision domain [7, 12, 14, 27]. These approaches follow a similar idea: pull together an anchor and a positive sample, meanwhile push apart the anchor from many negative samples. The positive sample is generated by a sets of data augmentations and the negative samples are randomly chosen from the mini-batch. The most widely used objective function is the InfoNEC [27], which encourages the mutual information between positive samples to be large while for negative samples, to be small. Recently, Khosla *et al*. [18] proposed a powerful contrastive loss that allows for multiple positives per anchor and proved its superior over traditional cross entropy under supervised setting. We borrow the similarity idea and propose the instance contrastive loss to effectively learn the instance embedding from image annotations.

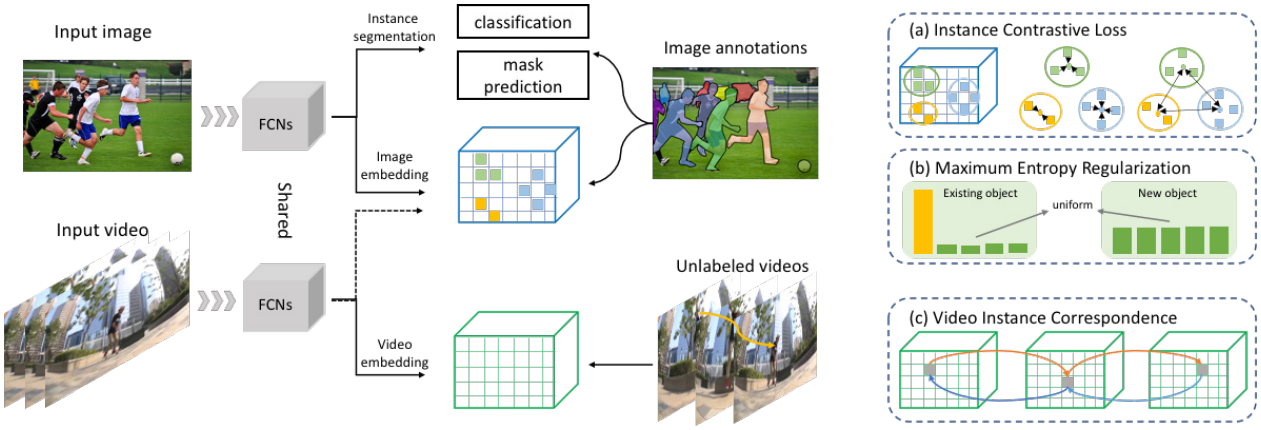

Figure 2. An overview of our proposed framework, which is built upon the bottom-up instance segmentation, *i.e.*, classification and mask prediction heads. We propose image/video embedding heads. We train the image embedding branch with (a) an instance contrastive loss; (b) a maximum entropy regularization term using image annotations only; and train the video embedding branch via (c) self-supervised video correspondence learning. See Sec. 3.2, 3.3, 3.4 for more details.

Self-supervised Learning in Videos aims to learn video-level representation by exploiting the frame redundancy. Some early work focus on representation learning from frames chronological order [24, 9, 41]. For instance, Misra *et al*. [24] attempts to determine whether a sequence of frames from a video is placed in the correct temporal order, which can be used as a pretext task to improve some downstream tasks like action recognition. Besides, the colorization can be also treat as the supervision signal. Recently, several work [39, 20, 17] show that the cycle-consistency in time can be utilized as the supervisory signal for learning visual representations from video. The key idea is that: given any patch of an image at the first frame, then track it forward and backward, it should return its original position and the trajectory should be a circle. Different from the existing methods, the correspondence module in our framework focus on instance-level correspondence rather than pixel-level correspondence.

3. Proposed Method

We introduce our approach in this section. The overall framework is illustrated in Fig. 2, which is built upon the bottom-up instance segmentation framework: SOLO [40]. SOLO converts instance segmentation into two pixel-level classification tasks, *e.g.*, instance classification and instance mask prediction. Specifically, the input image is divided into $s \times s$ grids, and if the instance's center falls into a grid cell, that grid cell is responsible for the above two tasks. We integrate a head that learns the proposed tracking embedding into it. The whole framework can be trained jointly and perform both instance segmentation in each frame, as well as tracking between frames. In this section, we mainly focus on how to learn the instance embedding.

We define the problem in Sec. 3.1, and introduce how to utilize labeled images to learn a embedding for instance tracking through (i) an instance contrastive loss (IC) in Sec. 3.2, and (ii) a maximum entropy (ME) regularization term in Sec. 3.3. We further improve its performance with unlabeled videos, as discussed in Sec. 3.4.

3.1. Problem Definition

In semi-supervised tracking, we have a labeled image dataset $\{X_{\text{Img}}, Y_{\text{Img}}\}$ where each individual image x^i_{Img} has its corresponding instance-level annotation y^i_{Img}, including an instance category, a location (provided by a bounding box or a keypoint), and a mask. Meanwhile, we also have an another video dataset $\{X_{\text{Vid}}\}$ where no videos are annotated. The goal of semi-supervised tracking is to learn a feature representation that can effectively associate instances in $\{X_{\text{Vid}}\}$ by only using the supervised information present in the image dataset.

3.2. Instance Contrastive Loss

To learn a feature representation capable of tracking, we want to ensure that it is (i) discriminative of different instances, and (ii) consistent regardless of the variations present in videos. In addition, the feature representation should (iii) focus more on appearance rather than location, since objects can move in time. Normally, such a feature embedding can be learned, *e.g.*, via a side branch trained with labelled identities across frames as the supervision signal, as is evidenced in several existing works [5, 16, 43]. Although no such annotations are accessible here, we find that instance-level annotation on images already provides sufficient information to achieve the above goals, *i.e.*, to distinguish which pixels belong to the same instance, and which are from different ones. In the following, we propose to learn this via a contrastive learning framework.

We illustrate our network architecture in Fig. 2: Other than the original classification and mask prediction heads in

SOLO [40], we integrate our embedding network for tracking in parallel with them, as a third head. We equip it with the same sub-network structure and feature map resolution as the classification head at each level in FPN [21] in order to make the network efficient and light-weight. We denote by $h(\cdot)$ the tracking head's mapping of the bottleneck representation to the tracking embedding, and by f the output feature map. We utilized the same grid-level instance labels that are assigned to the classification branch in SOLO and in several other works [19, 35, 40]: On the ground truth instance label images, we regard one pixel (x, y) to belong to one instance if it falls into a range, controlled by scale factors $\varepsilon : (cx, cy, \varepsilon w, \varepsilon h)$, where (cx, cy), w, and h denote the center of mass, width and height of the given ground truth mask. The instance assignment maps are down-sampled and rounded to fit the resolution of each level. More details can be found in [40]. Similar to the classification head, the feature map is much smaller in size than the original image, e.g., 40×40 at the most fine-grain level. We refer to each element as a grid cell.

With grid-level instance labels, we can directly extend the original formulation of contrastive learning [14, 34], based on InfoNCE [27] to the instances of each image. With slight abuse of notation, for one query grid cell $x_q \in X$ with feature f_q from the i^{th} instance Ω_i, we sample another vector f_p from the same instance as the positive sample, and all the other grid cells from different instance as the negative ones. We thus optimize for the pixel x_q:

$$\mathcal{L}_q = -\log \frac{\exp(f_p^\top \cdot f_q)}{\sum_{k \in \Omega_{\bar{i}}} \exp(f_k^\top \cdot f_q)}, \quad p, q \in \Omega_i \tag{1}$$

where $\Omega_{\bar{i}}$ is the set of cells from all the other instances $\bar{i}$. However, we found that (1) does not perform well in our case due to the highly long-tailed distribution of instances w.r.t. to their number of pixels. *E.g.*, smaller instances will be insufficiently trained due to less positive samples.

Center-Contra Losses. We address the above issue by proposing a novel form of the loss: a combination of center and contrastive (Center-Contra) losses. We obtain the center representation C_i of an instance i by averaging all embedding features assigned with this instance, as $C_i = \frac{1}{N_i} \sum_{q \in \Omega_i} f_q$. Here N_i represent the number of grid cells in Ω_i. To force the embedding feature vectors of the same instance to be similar, we introduce the center loss that minimizes the L1 distance:

$$\mathcal{L}_i^{\text{center}} = \sum_{q \in \Omega_i} \|C_i - f_q\|_1. \tag{2}$$

Meanwhile, the embedding of different instances also need to be distinct from each others in order for the embedding to have a strong discriminative ability. Thus, we propose a contrast term by pushing the center representation of all the instances $\{C_i | i \in [1, K]\}$ further apart, where

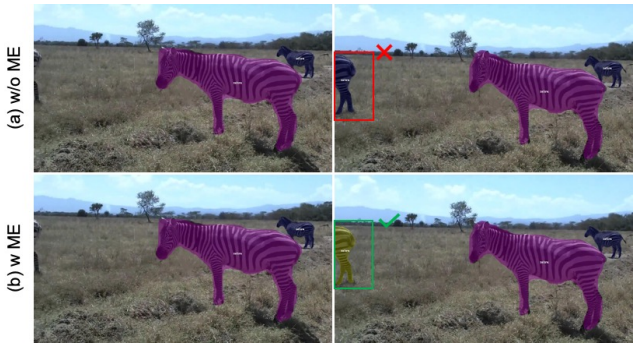

Figure 3. An illustration of failure case when a new object appears and the effectiveness of maximum entropy (ME) regularization. Row (a) and (b) are results without and with ME regularization. Best viewed in color and zoom in to see details.

K is the number of instances in an image. In particular, we compute a dense similarity matrix:

$$S(i, j) = \frac{\exp(C_i^\top \cdot C_j)}{\sum_{k=0}^{K} \exp(C_i^\top \cdot C_k)}, \tag{3}$$

To push apart instances, we need to encourage the elements on the diagonal of the matrix $S_{i,i}$ to be larger than the other off-diagonal elements $S_{i,j}, \forall j \neq i$. Thus, we maximize the self-matching likelihoods, where **CE** is the cross-entropy loss and I is the identity matrix:

$$\mathcal{L}^{\text{contra}} = \mathbf{CE}(S, I). \tag{4}$$

Finally, we enforce IC losses by summing up the center losses of all instances, and combining them with the contrast term:

$$\mathcal{L}^{\text{IC}} = \sum_{i=0}^{K} \mathcal{L}_i^{\text{center}} + \lambda \mathcal{L}^{\text{contra}}. \tag{5}$$

Compared to utilizing individual feature vectors, contrastive loss based on the center embedding in (4) effectively avoids the issue of highly-imbalanced size of instances.

Tracking an Instance via the Embedding. Given the learned embedding for tracking, we utilize the $\{C_i | i \in [1, K]\}$ as the prototype representations of instances to perform tracking, *i.e.*, grid cells of the next frame are directly classified into K classes by comparing against these prototypes through a softmax function, where the classification score indicates the instance associations. In addition, tracking can also be improved by leveraging information from the classification prediction branch, which is further discussed in Sec. 4.1.

3.3. Maximum Entropy Regularization

So far, our tracking approach is based on the assumption that any instance in the current frame also exists in the previous frame. It doesn't consider newly emerged objects. We observe that with the tracking procedure described in

Sec. 3.2, a new object is highly likely to exhibit a peaky distribution for its similarity score when matched to all the instances in the previous frame. Consequently it will be incorrectly matched to an existing instance, *e.g.*, see the dark black zebra in Fig. 3, top.

To resolve this issue, we apply entropy maximization so that the model performs out-of-distribution detection [36] – which means ideally, a new object should not bear more resemblance to any of one existing instances in comparison to the others. Since we do not have video labels that annotate new objects in time, we exploit the existing image's labels by adding a ME term for all the instances in it: we increase the entropy measured for the similarity between the center embedding of each instance and all other instances. Reusing the similarity matrix S, the entropy is computed as:

$$H = -\sum_{i}^{K}\sum_{j\neq i}^{K} S(i,j)\log(S(i,j)), \tag{6}$$

where K is the number of instances and $S(i,j)$ is the probability of matching instance i to j. High entropy H indicates uniform output probability. When enforced together with the IC term (5), it encourages instances to be equally dissimilar to all other instances, see Fig. 2 (b).

When a new object is successfully detected, we follow the tracking strategy described in the previous section by comparing it to the existing K objects (already detected in previous frames). Via ME, we enforce the similarity scores to be equally low for all existing instances as shown in Fig. 2 (b). Thus, it is easy to assign a new identity to a new object by setting a proper threshold such that all similarity scores are below it. Fig. 3 shows a comparison of the model without and with the proposed ME term.

3.4. Self-supervised Video Correspondence

Although large-scale videos are hard to label, they are easy to acquire. Can we further improve our model by leveraging these videos? The answer is positive, but non-trivial: On the one hand, with a tracking embedding trained only with image collections, there is no guarantee that tracking of instances can be continuous and coherent over time. However, with videos we do not know the ground truth instance correspondences. Moreover, with videos we also need to address the domain gap that usually exists between image and videos.

To this end, we leverage self-supervised video correspondence learning [17, 20, 39] to regularize tracking of the predicted instances. We determine the valid grid cells (*i.e.*, those belonging to any instances) through non-maxima suppression (NMS) on the matches with higher classification response (see inference in [40]) for more details). On the tracking embedding, we learn grid cell-level video correspondences in the valid grid cells only, *i.e.*, within the regions containing instances, through a cycle consistency loss [17, 39]. In detail, given a group of frames randomly sampled from one sequence, we compute cross-instance affinity $A \in \mathbb{R}^{P\times Q}$, where P, Q are the numbers of valid instances in a pair of frames. Let $A_t^{t+1}(i,j)$ be the transition probability of the i^{th} instance at time t being matched with the j^{th} instance at time $t+1$. We can formulate long-range correspondences by the chain rule:

$$\bar{A}_t^{t+k} = \prod_{i=0}^{k-1} A_{t+i}^{t+i+1}. \tag{7}$$

If we reverse this sequence and track the instances from $t+k$ to t, ideally, the i^{th} instance should return back to its original position in the first frame. Thus we have the following objective, where I is the identity matrix:

$$\mathcal{L}^{\text{cyc}} = \mathbf{CE}(\bar{A}_t^{t+k}\bar{A}_{t+k}^{t}, I). \tag{8}$$

We note that differently from [17], which needs to maintain a group of large affinity matrices (*i.e.*, $N\times N$ where N is the number of pixels), the dimensions of affinity in our case (*i.e.*, number of valid grids) is much smaller and the module is more efficient.

In addition, we observe that when a domain gap between image and video datasets exists, *e.g.*, COCO [22] vs YouTube-VIS [43], adopting the video objective (8) on the tracking embedding does not ensure convergence due the shared normalization. Therefore, we instead learn a video embedding using (8) with an additional head (see Fig. 2, the dashed link is not used when domain gap exists). We found that with a shared backbone network, both the image embedding and the video embedding can be improved by self-supervised learning. During inference, we utilize the image embedding for tracking due to its superior performance.

3.5. Test-time Adaptation

Inspired by [32], we can further mitigate the distribution shifts during the test-time: We still adopt the video embedding branch, and update the model weights by keeping the video correspondence loss in an online adaptation fashion. We find that the best performance can be achieved by updating the weights from the viedo correspondence branch as well as the backbone network (including the FPN Head [21]).

4. Experiments

We evaluate our proposed method on two different instance-level tracking problems: video instance segmentation and multi-person pose tracking.

4.1. Datasets and Evaluation Metrics

YouTube-VIS [43] is the first and largest dataset for video instance segmentation. In each video, objects with bounding boxes and masks are labeled manually every five

Methods	Video Annotations	With Embed	Contrastive Loss	Max Entropy	Video Correspondence	AP	$AP_{0.5}$	$AP_{0.75}$	AR_1	AR_{10}
MaskTrack-RCNN [43]	✓	✓				29.0	47.5	32.2	28.7	32.4
SOLO [40]						23.9	43.3	21.5	26.7	37.3
SOLO-Track		✓	✓			28.4	50.0	30.4	27.6	34.4
SOLO Track		✓	✓	✓		29.7	52.8	29.9	30.7	34.9
SOLO-Track		✓	✓	✓	✓	**32.9**	**54.4**	**35.0**	**34.1**	**40.8**

Table 1. Ablation study with different proposed components on YouTube-VIS validation set. The best results are highlighted in bold.

# frames	AP	$AP_{0.5}$	$AP_{0.75}$	AR_1	AR_{10}
2	**32.9**	**54.4**	**35.0**	**34.1**	**40.8**
3	31.8	52.4	31.7	32.2	39.1
4	30.9	51.6	30.9	31.7	38.4

Table 2. The performance of video instance segmentation with different number of frames in video correspondence model. The best results are highlighted in bold.

frames and the identities cross different frames are annotated as well. Since only the validation set is available for evaluation, all results reported in this paper are evaluated on the validation set. It is important to note that for VIS, we only test on the videos whose categories overlap with COCO [22], which are 20 categories. We contacted the authors for the annotations of that sub validation set.

PoseTrack [2] is a large-scale benchmark for multi-person pose estimation and tracking. It contains challenging sequences of people in dense crowds performing a wide range of activities. We conduct experiments only on PoseTrack 2018, where each person is annotated with 15 body joints, each one defined as a point and associated with a unique person id cross frames.

Evaluation Metrics. For VIS, we use the metrics mentioned in [43], which are average precision (AP) and average recall (AR) based on a spatio-temporal Intersection-over-Union (IoU) metric. For pose tracking, we evaluate our model via standard pose estimation [30] and tracking metrics [2], which are expressed by AP and multi-object tracking accuracy (MOTA), respectively. Unlike [11, 32, 37], we report MOTA along with its corresponding AP after post-processing videos. We apply post-processing to ignore some keypoints that are below a predefined confidence score. Note that it can lower the performance on AP but improve the performance on MOTA.

4.2. Implementation Details

Training. For both VIS and pose tracking, we first pretrain our model on the COCO dataset with the instance embedding head with the IC loss and ME regularization. In particular, we utilize SOLO and PointSetAnchor [42] as the base models for instance segmentation and pose estimation, respectively. The details of instance and keypoint embedding modules are described in supplementary materials. Our model is implemented on MMDetection [6] and the whole framework is trained with 8 NVIDIA TITAN V100 GPUs until convergence.

Inference. During evaluation, the testing video is processed frame by frame in an online fashion as described in [43]. More details can be found in supplementary materials. To keep consistent with the previous approaches and improve the performance, we also apply a post-processing procedure introduced in [43], which combines the initial prediction results with: detection confidence, bounding box IoU, category consistency, and similarity scores, etc. During the test-time training, each video is finetuned for 5 iterations with the same hyper-parameters as the training.

4.3. Ablation Study

We conduct all ablation studies on the YouTube-VIS dataset. We believe that similar conclusions can also be drawn for pose tracking.

Baseline Model. To the best of our knowledge, this is the first work to learn semi-supervised tracking using only image annotations, and hence it is important to establish a strong baseline model. In particular, we use MaskTrack RCNN [43] as the fully supervised baseline. It takes the pretrained MaskRCNN model and finetunes it on YouTube-VIS [43] with full video annotations, including instance categories, locations, masks and identities. The MaskTrack baseline is used to show how well our proposed semi-supervised method performs compared to fully supervised state-of-the-art methods. In addition, we also provide a bottom-up baseline based on SOLO, by training the task of instance segmentation without learning the tracking embedding. The objects are associated by spatial distance and category consistency. It is clear that the SOLO baseline is less accurate than MaskTrack RCNN. The SOLO baseline is used to validate the effectiveness of each proposed component in our method.

Effectiveness of Instance Contrastive Loss. To validate its effectiveness, we report the performance with, and without the embedding branch in Table 1. With our proposed IC loss, the performance is improved by 4.5%, 6.7% and 8.9% in AP, $AP_{0.5}$ and $AP_{0.75}$, respectively compared to the SOLO baseline. This improvement validates the previous claim that even only trained with labeled images, our method can learn discriminative representation with strong tracking capability.

Effectiveness of Maximum Entropy Regularization. Besides strong distinguishing ability, a robust embedding also needs to discover new objects. However, as shown in Fig 3, the embedding feature cannot distinguish between new and existing objects effectively by only using the IC loss. Thus, the maximum entropy (ME) regularization term

is proposed to address this problem. As listed in Table 1, the model with the ME regularization term can effectively boost performance on VIS. Specifically, it improves AP and $AP_{0.5}$ by 1.3% and 2.8%, and it achieves an AP of 29.7%, which outperforms the fully supervised baseline of MaskTrack RCNN [43].

Effectiveness of Video Correspondence. We also show the effectiveness of self-supervisely learn video correspondence with unlabeled videos that are fairly cheap and easy to obtain. As listed in Table 1, the proposed video correspondence model can improve performance significantly across all evaluation metrics. For instance, the gains in AP, $AP_{0.5}$ and $AP_{0.75}$ are 2.8%, 1.6% and 5.11%, respectively. In addition, compared to the SOLO baseline, our final model improves the performance by 9.0%, 11.1% and 13.5% for AP, $AP_{0.5}$ and $AP_{0.75}$, respectively. Furthermore, it also outperforms MaskTrack RCNN by a large margin. These improvements show that the video correspondence model can significantly enhance the tracking capability of our embedding representation.

Sequence Length. So far we have validated all our proposed components. The video correspondence model especially brings a significant improvement, but the number of frames used to compute the cycle loss can affect its performance a lot. We can only perform the experiment with 2 to 4 frames due to limitations on GPU memory. From Table 2, it can be observed that the video correspondence model can achieve the best performance using only two frames. With increased number of frames, the performance on AP drops gradually from 32.9% to 30.9%. The degraded results may be caused by inclusion of noisy sampled with more frames. Since we do not have any annotations, we instead use category-level predictions to sample several positive instances. While the predictions are not exactly accurate, more frames can bring more noise, which leads to the worse performance.

4.4. Comparison with State-of-the-Art Methods

Video Instance Segmentation. Since we test on a subset of the YouTube-VIS validation set, we either evaluate the publicly released models, or our re-implementation of the other approaches. The comparison results are shown in Table 3. Both the MaskTrack RCNN and SipMask have a tracking branch to learn object embedding representation from labeled videos. Compared to them, our method, although does not involve any annotation of videos, can still achieve comparable performance. Furthermore, with the video instance correspondence module, our approach achieves the best performance across all evaluation metrics.

In addition, we compare our approach to the methods involving various cues for post processing. IoUTracker+ [4] assigns the instance label with the largest score to a candidate box. Since it does not leverage any visual information,

Methods	AP	$AP_{0.5}$	$AP_{0.75}$	AR_1	AR_{10}
Video + Image Annotations					
MaskTrack R-CNN [43]	29.0	47.5	32.2	28.7	32.4
SipMask [5]	24.1	42.0	26.0	26.2	28.6
Only Image Annotations					
Ours	29.7	52.8	29.9	30.7	34.9
Ours$^+$	**32.9**	**54.4**	**35.0**	**34.1**	**40.8**
After post-processing					
Video + Image Annotations					
IoUTracker+ [43]	29.4	48.5	30.6	32.1	34.2
SeqTracker [43]	31.8	52.2	35.8	32.2	34.4
MaskTrack R-CNN [43]	36.0	58.4	**40.2**	35.4	38.9
SipMask [5]	**37.7**	57.8	38.0	37.4	40.3
Only Image Annotations					
Ours	34.1	58.0	37.9	33.0	39.2
Ours$^+$	37.4	59.7	39.1	36.4	43.8
Ours*	**38.3**	**61.1**	39.8	36.9	**44.5**

Table 3. Comparison of the our approach with the SOTA methods on the YouTube-VIS validation set. "Ours" represents the model with instance embedding branch trained with IC loss and ME regularization. "Ours$^+$" stands for the model with the video correspondence module as well. "Our* is the model updated by test-time adaptation upon "Ours$^+$". The best results are highlighted in bold.

	Methods	MOTA					AP
		Head	Shou	Wrist	Ankle	Total	
	Video + Image Annotations						
Top down	Miracle [44]	68.8	73.5	61.2	56.7	64.0	–
Top down	OpenSVAI [25]	–	–	–	–	62.4	69.7
Top down	LightTrack [26]	–	–	–	–	64.6	72.4
Top down	KeyTrack [31]	–	–	–	–	66.6	74.3
Bottom up	MDPN [13]	50.9	55.5	49.0	45.1	50.6	71.7
Bottom up	STAF [29]	–	–	–	–	60.9	70.4
Bottom up	MIPAL++ [16]	76.0	76.9	56.4	52.4	65.7	74.6
	Only Image Annotations						
	Baseline	64.9	70.9	56.3	55.0	62.0	69.2
	Ours	65.8	71.6	56.3	56.6	62.8	69.3
	Ours$^+$	67.1	72.3	58.2	57.7	64.2	69.3
	Ours^{++}	70.4	73.3	55.9	56.3	64.7	71.4

Table 4. Comparison of our approach with the SOTA methods on the PoseTrack2018 validation set."Baseline" associates poses only by the OKS metrics. "Ours" and "Ours$^+$" have the same definitions as Table 3. "Ours^{++}" has the same structure as the "Ours$^+$" model, but is finetuned with the MPII data [3]. The best results on MOTA and AP for the methods with both image and video annotations and only image annotations are highlighted with red and blue color, respectively.

its performance is a little weaker. SeqTracker [43] first computes instance segmentation results for all frames of a video, and then searches all possible tracks to find the one with the largest score. MaskTrack RCNN and SipMask perform the post-processing proposed by [43] to have more comprehensive cues for object association. By adopting a similar post-processing strategy, our approach can achieve comparable or even better performance versus other SOTAs. Furthermore, with the help of self-supervised Test-time adaption strategy, we can improve the final performance by more than 1% on AP and $AP_{0.5}$. Fig. 4 (Row 1-2) shows some qualitative results on YouTube-VIS validation set. Each row represents the predicted results on different frames in a video.

Figure 4. Visualization results of our proposed semi-supervised tracking approach on video instance segmentation and pose tracking. Each row has five sampled frames from a video sequence. Categories, bounding boxes and instance masks are shown for each object. Note that objects with the same predicated identity across frames are marked with the same color. Zoom in to see details.

Analysis of Post-processing. We notice that similar post-processing steps bring much more improvement to methods that train with video annotations than our approach. For instance, the AP performance of MaskTrack R-CNN [43] and SipMask [5] improves by 7.0% and 13.6%, respectively, with category and spatial consistency. However, the improvement of our method is only about 4%. This is because post-processing takes additional cues from the instance segmentation results, *i.e*. category prediction, bounding box localization and mask prediction. However, due to the obvious domain gap between the training set of COCO, and the testing set of YouTube-VIS, the performance of both modules drops accordingly, and thus the limited improvement after post-processing compared to the others. We note we mainly focus on learning a tracking embedding representation in this work. We leave domain adaptation of the original SOLO heads to the further work.

Pose Tracking. Besides video instance segmentation, our approach can also be extended to human body pose tracking. We compare our approach with the SOTAs and report results on the validation set of PoseTrack2018. The results are summarized in Table 4. Note that since the number of joints and their definitions are different in COCO [22] and PoseTrack [2], an additional finetuning step on MPII [3] is employed (denoted as Our^{++} in Table 4). In general, our proposed method can achieve comparable results to both top-down and bottom-up methods. For instance, comparing with the top-down methods, although our performance on AP is slightly lower, our performance on MOTA is quite competitive. However, the top-down methods always detect the human body first and perform pose estimation and tracking on cropped person images, which are much slower than ours. The analysis of running time is included in the supplementary material. Additionally, our approach even outperforms most of bottom-up methods. For instance, compared to STAF [29], the improvement is substantial: +3.8% on MOTA and +1.0% on AP.

5. Conclusion

We introduce a novel semi-supervised framework that can achieve instance tracking without any video annotations. The Instance Contrastive loss and Maximum Entropy regularization are proposed to learn the discriminative representation of different instances capable of tracking via image annotations. Furthermore, in order to leverage the unlabeled videos, which are more accessible in the real-world, we propose to learn video correspondence in a self-supervised manner. Instead of learning a separated network, we integrate all proposed components into existing bottom-up instance segmentation or pose tracking frameworks. Extensive experiments demonstrate that our proposed method performs on par if not better than most STOA approaches.

References

[1] The 2nd large-scale video object segmentation challenge. https://youtube-vos.org/challenge/2019/. Accessed: 2019-11-12. 2

[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 *Conf. Comput. Vis. Pattern Recog.*, pages 5167–5176, 2018. 6, 8

[3] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Conf. Comput. Vis. Pattern Recog.*, pages 3686–3693, 2014. 7, 8

[4] Erik Bochinski, Volker Eiselein, and Thomas Sikora. High-speed tracking-by-detection without using image information. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE, 2017. 7

[5] Jiale Cao, Rao Muhammad Anwer, Hisham Cholakkal, Fahad Shahbaz Khan, Yanwei Pang, and Ling Shao. Sipmask: Spatial information preservation for fast instance segmentation. *Eur. Conf. Comput. Vis.*, 2020. 1, 3, 7, 8

[6] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 6

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 2

[8] Minghui Dong, Jian Wang, Yuanyuan Huang, Dongdong Yu, Kai Su, Kaihui Zhou, Jie Shao, Shiping Wen, and Changhu Wang. Temporal feature augmented network for video instance segmentation. In *Int. Conf. Comput. Vis. Worksh.*, 2019. 1, 2

[9] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *Conf. Comput. Vis. Pattern Recog.*, pages 3636–3645, 2017. 3

[10] Yang Fu, Linjie Yang, Ding Liu, Thomas S Huang, and Humphrey Shi. Compfeat: Comprehensive feature aggregation for video instance segmentation. *AAAI*, 2020. 2

[11] Rohit Girdhar, Georgia Gkioxari, Lorenzo Torresani, Manohar Paluri, and Du Tran. Detect-and-track: Efficient pose estimation in videos. In *Conf. Comput. Vis. Pattern Recog.*, pages 350–359, 2018. 6

[12] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Adv. Neural Inform. Process. Syst.*, 33, 2020. 2

[13] 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 *Eur. Conf. Comput. Vis.*, 2018. 7

[14] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Conf. Comput. Vis. Pattern Recog.*, pages 9729–9738, 2020. 2, 4

[15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Int. Conf. Comput. Vis.*, 2017. 2

[16] Jihye Hwang, Jieun Lee, Sungheon Park, and Nojun Kwak. Pose estimator and tracker using temporal flow maps for limbs. In *Int. Joint Conf. Neural Network*, pages 1–8. IEEE, 2019. 3, 7

[17] Allan Jabri, Andrew Owens, and Alexei Efros. Space-time correspondence as a contrastive random walk. *Adv. Neural Inform. Process. Syst.*, 33, 2020. 1, 2, 3, 5

[18] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020. 2

[19] Tao Kong, Fuchun Sun, Huaping Liu, Yuning Jiang, Lei Li, and Jianbo Shi. Foveabox: Beyond anchor-based object detector. *IEEE Trans. Image Process.*, pages 7389–7398, 2020. 4

[20] Xueting Li, Sifei Liu, Shalini De Mello, Xiaolong Wang, Jan Kautz, and Ming-Hsuan Yang. Joint-task self-supervised learning for temporal correspondence. In *Adv. Neural Inform. Process. Syst.*, pages 318–328, 2019. 1, 3, 5

[21] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Conf. Comput. Vis. Pattern Recog.*, 2017. 4, 5

[22] 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 *Eur. Conf. Comput. Vis.*, 2014. 5, 6, 8

[23] Jonathon Luiten, Philip Torr, and Bastian Leibe. Video instance segmentation 2019: A winning approach for combined detection, segmentation, classification and tracking. In *Int. Conf. Comput. Vis. Worksh.*, 2019. 1, 2

[24] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *Eur. Conf. Comput. Vis.*, pages 527–544. Springer, 2016. 3

[25] Guanghan Ning, Ping Liu, Xiaochuan Fan, and Chi Zhang. A top-down approach to articulated human pose estimation and tracking. In *Eur. Conf. Comput. Vis.*, pages 0–0, 2018. 7

[26] Guanghan Ning, Jian Pei, and Heng Huang. Lighttrack: A generic framework for online top-down human pose tracking. In *Conf. Comput. Vis. Pattern Recog. Worksh.*, pages 1034–1035, 2020. 7

[27] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2, 4

[28] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017. 1

[29] Yaadhav Raaj, Haroon Idrees, Gines Hidalgo, and Yaser Sheikh. Efficient online multi-person 2d pose tracking with

recurrent spatio-temporal affinity fields. In *Conf. Comput. Vis. Pattern Recog.*, pages 4620–4628, 2019. 7, 8

[30] Matteo Ruggero Ronchi and Pietro Perona. Benchmarking and error diagnosis in multi-instance pose estimation. In *Int. Conf. Comput. Vis.*, pages 369–378, 2017. 6

[31] Michael Snower, Asim Kadav, Farley Lai, and Hans Peter Graf. 15 keypoints is all you need. In *Conf. Comput. Vis. Pattern Recog.*, pages 6738–6748, 2020. 7

[32] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Conf. Comput. Vis. Pattern Recog.*, pages 5693–5703, 2019. 5, 6

[33] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *Int. Conf. Machine Learn.*, pages 9229–9248. PMLR, 2020. 2

[34] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019. 4

[35] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Int. Conf. Comput. Vis.*, pages 9627–9636, 2019. 4

[36] Apoorv Vyas, Nataraj Jammalamadaka, Xia Zhu, Dipankar Das, Bharat Kaul, and Theodore L Willke. Out-of-distribution detection using an ensemble of self supervised leave-out classifiers. In *Eur. Conf. Comput. Vis.*, pages 550–564, 2018. 5

[37] Manchen Wang, Joseph Tighe, and Davide Modolo. Combining detection and tracking for human pose estimation in videos. In *Conf. Comput. Vis. Pattern Recog.*, pages 11088–11096, 2020. 6

[38] Qiang Wang, Yi He, Xiaoyun Yang, Zhao Yang, and Philip Torr. An empirical study of detection-based video instance segmentation. In *Int. Conf. Comput. Vis. Worksh.*, 2019. 2

[39] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *Conf. Comput. Vis. Pattern Recog.*, pages 2566–2576, 2019. 1, 3, 5

[40] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. Solo: Segmenting objects by locations. *Eur. Conf. Comput. Vis.*, 2020. 2, 3, 4, 5, 6

[41] Donglai Wei, Joseph J Lim, Andrew Zisserman, and William T Freeman. Learning and using the arrow of time. In *Conf. Comput. Vis. Pattern Recog.*, pages 8052–8060, 2018. 3

[42] Fangyun Wei, Xiao Sun, Hongyang Li, Jingdong Wang, and Stephen Lin. Point-set anchors for object detection, instance segmentation and pose estimation. *Eur. Conf. Comput. Vis.*, 2020. 2, 6

[43] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *Int. Conf. Comput. Vis.*, 2019. 1, 2, 3, 5, 6, 7, 8

[44] Dongdong Yu, Kai Su, Jia Sun, and Changhu Wang. Multi-person pose estimation for pose tracking with enhanced cascaded pyramid network. In *Eur. Conf. Comput. Vis.*, pages 0–0, 2018. 7