

Learning to Track Any Object

Achal Dave
CMU

Pavel Tokmakov
CMU

Cordelia Schmid
Google Research

Deva Ramanan
CMU

Abstract

Object tracking can be formulated as “finding the right object in a video”. We observe that recent approaches for class-agnostic tracking tend to focus on the “finding” part, but largely overlook the “object” part of the task, essentially doing a template matching over a frame in a sliding window. In contrast, class-specific trackers heavily rely on object priors in the form of category-specific object detectors. In this work, we repurpose category-specific appearance models into a generic objectness prior. Our approach converts a category-specific object detector into a category-agnostic, object-specific detector (i.e. a tracker) efficiently, on the fly. Moreover, at test time the same network can be applied to detection and tracking, resulting in a unified approach for the two tasks. We achieve state-of-the-art results on two recent large-scale tracking benchmarks (OxUvA and GOT, using external data). By simply adding a mask prediction branch, our approach is able to produce instance segmentation masks for the tracked object. Despite only using box-level information on the first frame, our method outputs high-quality masks, as evaluated on the DAVIS ’17 video object segmentation benchmark.

1. Introduction

Tracking is an essential element of video analysis. Extracting spatio-temporal regions corresponding to objects from a video is not only the end goal for surveillance and video labeling [19], but also an important intermediate representation for tasks such as action recognition [44, 51].

Unfortunately, tracking in general is notoriously challenging and potentially ambiguous. Consider the example in Figure 1 of tracking a bus with only one side visible initially. Without prior knowledge, it is unclear whether the back side of the bus (visible in future frames) is a different viewpoint of the same bus, or a new object itself. In practice, many tracking approaches struggle to resolve such ambiguities, tending to diverge to an object part which is most similar to the initial template (e.g., the back window of the bus). Successful tracking in these scenarios necessitates *object priors*. Indeed, approaches for category-specific



Our approach

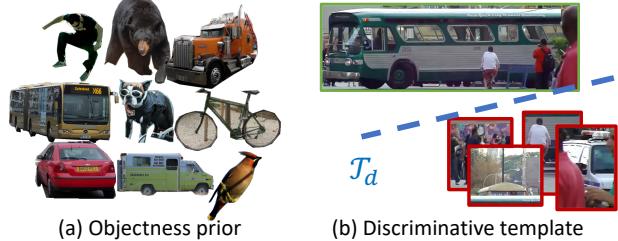


Figure 1: Objects of interest in generic, user-initialized tracking share a common set of *objectness* traits. Our approach (a) learns a generic objectness prior from image-based datasets, and (b) adapts it to a specific object of interest (e.g. the bus in the top left) by computing a linear discriminator between the object and its background in closed form. This allows tracking objects through significant deformations without latching onto distractors.

tracking, where the tracked object categories are known before hand, heavily rely on priors in the form of category-specific object detectors [2, 48, 46, 5]. By contrast, approaches for user-initialized tracking have largely eschewed such priors [7, 6, 53] in the pursuit of tracking generic objects, sometimes known as model-free tracking [47, 22]. However, generic objects still share a common set of *objectness* traits [1]. How can we operationalize this implicit constraint into a useful prior?

In this work, we repurpose category-specific appearance models into a generic *objectness* prior that can be used for category-*agnostic* tracking. In essence, we show that model-free tracking is far easier with better models! Doing so requires tackling two key challenges, shown in Figure 1: (1) How do we best adapt a category specific prior into a

generic objectness prior? (2) How do we further adapt this generic prior to the particular instance of interest?

To address (1), we build a joint model for category-specific object detection and category-agnostic tracking (Figure 2). It is based on the Mask R-CNN [15] object detection architecture. For tracking, it takes as an additional input an object template in the first frame and computes its feature embedding. This template is then used to compute the similarity between the object of interest and a new frame. The similarity map is in turn applied to reweight spatial features from the new frame to detect only the object of interest. Importantly, training the network jointly on image and video datasets, allows us to both capture a generic object appearance model from the diverse image data and learn to use it in a category-agnostic way for tracking.

To address (2) – e.g., better separating the bus in Figure 1 from other vehicles, such as the van on the right – we propose a lightweight on-the-fly adaptation strategy. We compute a linear separator (\mathcal{T}_d in Figure 1) between the object of interest and other objects in the first frame. This separator is computed in closed form in a fully differentiable manner, and applied in future frames to compute similarities.

An intriguing property of our proposed architecture is that it can be used both as a single-object tracker and an object detector. Moreover, by capitalizing on the mask prediction branch of [15], we are able to train and test the same network for instance and video object segmentation. To sum up, we present a single unified approach for object detection, tracking, instance and video object segmentation.

We evaluate our model on two recent, large scale datasets for object tracking: OxUvA [40] and GOT [18]. The former is focused on long-term object tracking, with objects undergoing a lot of appearance variation and occlusion. In contrast, the videos in GOT are shorter, but contain diverse object categories, covering more than 560 object classes. Our method outperforms prior work on OxUvA, and outperforms state-of-the-art approaches that use external data on GOT by a large margin. Next, we show results competitive with the state-of-the-art on the LTB-35 dataset from the VOT 2018 Long Term challenge [21]. Finally, we validate the quality of our masks on DAVIS’17 dataset for video segmentation [32], demonstrating that our unified approach performs on par with specialized video segmentation methods that don’t finetune on the test videos.

Our contributions are three-fold: (1) we incorporate an objectness prior in a generic tracker with a joint model for object detection, tracking, instance and video object segmentation; (2) we propose a lightweight strategy for computing discriminative object templates in an end-to-end fashion for efficiently handling distractors; (3) our method demonstrates state-of-the-art results on three benchmark datasets for object tracking and video object segmentation.

2. Related work

Single object tracking. Classical approaches for single object tracking, which requires tracking an object given a bounding box annotation in the first frame, were based on the tracking-by-detection paradigm: in many cases the detector is used to first to localize all the objects in a frame. The box corresponding to the object of interest was then selected by a discriminative classifier trained on the first frame annotation [3, 17, 20]. Correlation filters were commonly used for classification due to their efficiency [7]. To address appearance variation, some models updated the object template over time [17, 36]. Recent approaches learn correlation filters on top of deep features [11, 39].

Current methods for tracking largely ignore the objectness prior provided by detectors. Instead, they rely on a Siamese network architecture (initially introduced for signature verification [8]) adapted for tracking [6, 16, 37].

Recently, there have been several attempts to introduce ideas from CNN-based detection architectures into Siamese trackers. In particular, Li et al. [24] use the similarity map obtained by matching the object template to the test frame as input to an RPN-like module adapted from Faster R-CNN [34]. Later this architecture was extended by introducing hard negative mining and template updating [53], adding a mask prediction branch [43], and using deeper models [23]. Our approach differs in that instead of integrating components of object detectors into a tracking pipeline in a heuristic way, we turn a state-of-the-art object detection framework into a tracker. This allows our model to fully utilize the objectness prior learned on COCO, outperforming the heuristic-based approaches significantly.

Video object segmentation. Methods for video object segmentation take a precise object mask as input in the first frame and output pixel-level segmentations for the object in each frame. Early methods for this task were based on mask propagation through a graph connecting superpixels in the neighboring frames [38, 45]. More recently, these methods have been outperformed by deep-learning based approaches, which capitalize on the success of image segmentation architectures [9, 31]. In particular, they fine-tune a model trained for foreground-background segmentation using the annotation in the first frame and evaluate it on the remaining frames of the video. Some approaches also update the model using its own predictions to handle appearance variation [42]. While these methods demonstrate impressive accuracy, they remain slow due to the need to update the model during evaluation. Alternative approaches, that do not require network fine-tuning have been proposed recently [10, 50, 41], but remain inferior in performance.

These approaches treat video object segmentation as a problem *independent* from object tracking, with the re-

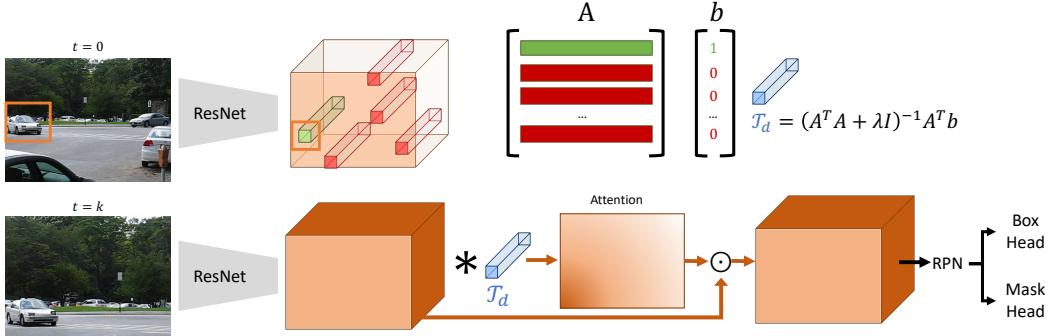


Figure 2: Overview of our approach. First, we use a state-of-the-art object detector [15] to extract features for the template image containing the object to be tracked (top left). Next, we compute a discriminative template that separates the features corresponding to the tracked object from the distractors in the first frame using linear regression (top right). Finally, attention masks computed with this template are used to reweight the feature maps of the detector to focus on the object of interest (bottom). Note that unlike standard, category-specific detectors, our box-head and mask-head output a single, category-agnostic prediction for the tracked object.

cent exception of [43]. In contrast, we adapt the intuition from [15] that instance masks can be computed as a by-product of object detection. Our tracker with a mask prediction branch achieves competitive performance on DAVIS’17 video object segmentation benchmark without requiring mask-level supervision on the first frame.

Object detection. CNN architectures for detection have brought significant progress, replacing classical methods for object detection that relied on hand-crafted features and part-based models [12]. Early approaches [13, 14] trained CNNs to classify pre-computed object proposals. More recent approaches solve the detection problem in an end-to-end way [27, 33, 34]. In particular, RCNN-like architectures [15, 34] operate in a two-stage fashion: first an RPN proposes a set of boxes, and pools features from each box region. Next, separate branches classify the object and refine the box coordinates. [25] introduced feature pyramid network (FPN) to aggregate features from several network layers. Finally, Mask R-CNN [15] extended this model to instance segmentation by adding a mask prediction branch. In this work, we convert this architecture into an object tracker by introducing a lightweight discriminative template matching block before the RPN. The resulting attention map guides the RPN to propose only boxes corresponding to the object of interest. Disabling the matching component turns the model into a standard object detector.

3. Method

An ideal model for tracking by detection can be described as a generic object detector that can be efficiently adapted to detect a specific object in a specific scene. In this section, we propose such an approach, shown in Figure 2. Our model leverages advances in standard object detection architectures by progressively incorporating modifications to build a state-of-the-art tracker, while maintaining the model’s detection capabilities.

We begin by briefly describing the Mask R-CNN architecture in Section 3.1. We then discuss our strategy of incorporating Siamese-like template matching into this model in a principled way in Section 3.2. Next, we propose our discriminative templates that efficiently integrate information about the distractors in Section 3.4. Finally, we discuss our strategy for training the unified model on object detection, tracking, and video segmentation datasets in Section 3.5.

3.1. Preliminaries

A Mask R-CNN detector, shown in Figure 2, consists of a backbone network (often a ResNet), a Region Proposal Network, and bounding box classification, regression and mask prediction heads. The former takes a frame as input and outputs a set of feature maps $\{C_1, C_2, C_3, C_4, C_5\}$, extracted from the respective blocks of the backbone and encoding the image with different degrees of spatial and semantic granularity. In practice, the output of the first block is discarded, due to memory constraints. The remaining feature maps are then updated via top-down lateral connections to propagate the information for the coarse but semantically rich top layers to the more spatially precise bottom layers, resulting in the final set of feature maps $\{P_2, P_3, P_4, P_5\}$ (see [25] for details). The feature dimensionality of these maps is fixed to 256, but their spatial dimensions decrease from fine to coarse, thus the resulting architecture is referred to as Feature Pyramid Network (FPN).

An RPN is implemented as a 3×3 convolutional layer that is applied to each FPN level in a sliding window fash-

ion, outputting an objectness score for each of the anchor boxes centered at the corresponding location. Crucially, the anchor boxes only capture various aspect ratios of the boxes, whereas scale variation is handled by the FPN. That is, a $1 \times 1 \times D$ dimensional feature at location (x, y) in P_5 represents the largest possible object centered at that location, whereas a feature of the same dimension at the corresponding location in P_2 represents the smallest possible object in centered in the same region. We use this observation to derive our scale-invariant object template in Section 3.2.

Finally, the top k boxes according to the RPN score are selected, and an ROI-Pool operation is used to convert their feature representations to a fixed size. The resulting features are passed to separate bounding box classification, regression, and mask prediction branches (see [15] for details). We now describe our approach to efficiently adapting this architecture to the task of object tracking.

3.2. Tracking as generalized object detection

Given a bounding box around the object of interest in the first frame, how can we adapt the Mask R-CNN detector to only track that specific instance? We take inspiration from Siamese-based approaches for tracking that store an object template from the first frame and compute a similarity between the template and the test frame representation in a sliding-window fashion. Differently from those methods, instead of localizing the objects directly via template matching, we use the resulting similarity map to reweight the feature representation of an object detector. This allows us to reuse the rest of the detection architecture and train the model jointly on images and videos.

Our key observation is that every object can be represented by a $1 \times 1 \times D$ feature \mathcal{T} in one FPN layer, corresponding to its scale and center location. Thus, we begin by extracting the corresponding representation for the template box in the first frame. In the standard detection training setup, Mask R-CNN assigns each groundtruth box in an image to a specific level in the feature pyramid, adding a loss that enforces that features at that scale generate a proposal around the groundtruth box. We use this same mapping to map our template box to the corresponding FPN level, and use the feature in that level that corresponds to the center of the template box. At test time, however, the scale of the object might have changed. Conveniently, we do not need to update the template to account for this, since scale variation is already handled by the FPN. Thus, we simply compute the similarity maps at all the levels of the feature pyramid via $S_i = P_i \star \mathcal{T}$, where \star stands for cross-correlation.

Next, instead of directly using the resulting similarities to localize the object, we propose to instead treat them as attention maps to guide the detector. To this end, we update the original FPN representations via $P_i \leftarrow P_i \cdot S_i$, where \cdot stands for the dot product. Notice that this operation simply

reweights the original representation, preserving the information used by the RPN in the next stage. Thus, we can naturally capitalize on the strong objectness prior learned by detectors on COCO, as well as learn to produce object masks for free. This re-weighted feature representation is used to generate and pool features for region proposals. The pooled, re-weighted features are finally passed through class-agnostic bounding box and mask regression heads.

At test time, our model produces multiple detections with confidence scores at every video frame. By default, we select the highest-scoring detection to construct the track, but we can make use of multiple detections by re-ranking them with external cues, such as predictions of an object dynamics model, or temporal smoothness cues (Section 4.2).

3.3. Joint Detection and Tracking

The modifications described above convert a standard Mask R-CNN detector into a tracker, which can not directly be used as a standard detector. We present two modifications that allow the tracker to be trained and evaluated as a standard, image-based detection model. First, when applied to a single image, we disable the attention module. Equivalently, this can be thought of as setting the attention to a uniform value of 1 at all pixel locations. Second, in order to output a class-specific bounding box and mask, as in standard Mask R-CNN, we instantiate a separate final layer for the box and mask regression heads for detection. Note that our model shares all parameters for detection and tracking *except* this final fully connected layer. We show in Section 4.3 that training jointly for detection with tracking improves tracking accuracy, while allowing our model to operate as a powerful single-frame detector, which can be useful for identifying distractor objects during tracking.

3.4. Discriminative Templates

Consider the frames from one of the videos in GOT shown in Figure 3 together with the corresponding similarity map S_i from the appropriate FPN level of our model. The model is supposed to track the cart in this video, however, the similarity map for the test frame shown in the bottom left is not localized on the object. We now propose a simple and efficient way of learning a discriminative template, increasing the robustness of the tracker.

Recall that in the FPN a feature vector at each location encodes an object centered at that region at the corresponding scale. Thus, sampling a large enough pool of features from all the levels outside of the ground truth bounding box naturally provides us with a training set for learning a linear discriminator for the object of interest in a given video. Moreover, such a discriminator can be found efficiently in a closed form via least squares. In particular, given a template \mathcal{T} and a set of negatives $N = \{\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_q\}$, we define

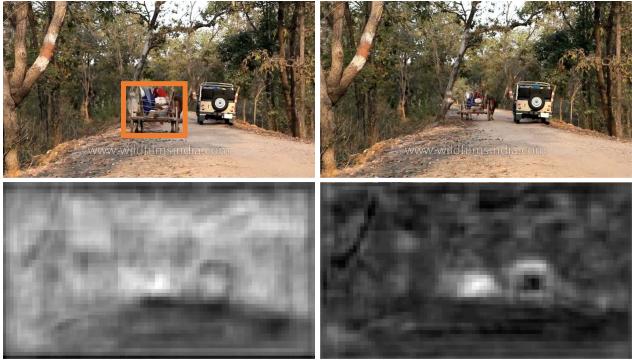


Figure 3: The effect of our proposed discriminative template on an example of a video from GOT-10k dataset [18]. By simply using the center feature of the bounding box around the cart (top left), the resulting attention map (bottom left) for the test frame (top right) is not focused on the object. In contrast, our discriminative template (bottom right) results in a much better attention map.

the data matrix A , and the label vector \mathbf{y} as follows:

$$A = [\mathcal{T}; \mathbf{n}_1; \mathbf{n}_2; \dots; \mathbf{n}_q], \mathbf{y} = [1; 0; 0; \dots; 0] \quad (1)$$

We then want to find a vector \mathcal{T}_d , which we call a discriminative template, that minimizes

$$\|A\mathcal{T}_d - \mathbf{y}\|_2^2 \quad (2)$$

holds. A closed form solution is available via:

$$\mathcal{T}_d = (A^T A + \lambda I)^{-1} A^T \mathbf{y}, \quad (3)$$

where I is the identity matrix and λ is a regularization hyper-parameter. We then use \mathcal{T}_d to compute the similarity maps in the same way: $S_i = P_i \star \mathcal{T}_d$.

Note that computing \mathcal{T}_d requires only a matrix inverse and matrix multiplications, operations which are fully differentiable in the elements of A , and can be implemented in standard deep learning frameworks. Thus, we can back-propagate through this computation. This guides the backbone to learn a feature space where objects can be separated via a linear classifier in an end-to-end manner.

Figure 3 (bottom right) shows that our discriminative template indeed significantly increases the precision of the similarity maps by incorporating the information about the distractors in a principled way. We now describe how we train our unified framework on dataset for object detection, tracking, and video segmentation.

3.5. Training

We first train our model on COCO for object detection [26] following Mask R-CNN training [15]. We then transfer the learned objectness prior to the tracking task.

To this end we add the discriminative template computation, and attention reweighting components described above, and fine-tune it on the ImageNet VID [35] and YTVOS [49] datasets. As ImageNet Vid does not provide segmentation groundtruth, we do not use it to update the mask branch. When fine-tuning for tracking, we make three simple modifications: (1) Our training batches consist of pairs of frames: for a batch of size K , we sample K videos, and then sample a *template* frame and a *search* frame at random from the video¹; (2) We re-weight FPN features of the search frame using the feature corresponding to the template frame’s bounding box; (3) Only a single, class-agnostic groundtruth box is used for training, which is the one corresponding to the tracked object in the search frame. These minor modifications allow us to maximally preserve the objectness priors learned on COCO.

4. Experiments

We begin with introducing the datasets used to train and evaluate our model, and providing the implementation details. Next we analyze the various choices made while designing our approach in Section 4.3. Finally, we compare our method to the state-of-the-art in Section 4.4.

4.1. Datasets and evaluation

We use the COCO [26] dataset to train our model for object detection, and ImageNet VID and YTVOS [49] to train the tracking module. We evaluate on two very recent, large scale tracking benchmarks: OxUvA [40] for long term tracking and GOT [18] for tracking of diverse objects. In addition, we use the DAVIS’17 [32] dataset for video object segmentation to evaluate the quality of the masks produced by our tracker, and the LTB35 videos from the VOT 2018 long term challenge to benchmark [21] against prior submissions to the challenge. We describe each of these datasets in more detail in the supplementary material.

4.2. Implementation details

Network architecture and training We use the Mask R-CNN detection framework throughout our experiments. In particular, we use the ResNet-50 FPN backbone, which achieves a useful balance between accuracy and efficiency. Our final model is trained for detection on MS COCO, as described in Sec. 4.3 and for bounding-box tracking on ImageNet VID and YTVOS. We will release training and evaluation code along with trained models upon acceptance.

Temporal heuristics Prior tracking approaches rely heavily on temporal information to simplify the tracking problem.

¹While we could limit the template frame to be the first frame of the video (as at test time), this would drastically reduce the diversity of our frame pairs.

Det Init?	Joint Det Train?	\mathcal{T}_d	Smooth?	OxUvA AUC	GOT AO
✗	✗	center	✗	63.2	64.7
✓	✗	center	✗	64.9	68.4
✓	✓	center	✗	65.8	68.6
✓	✓	mean diff	✗	67.6	68.8
✓	✓	mean pos	✗	69.1	69.1
✓	✓	lin. reg.	✗	71.1	69.5
✓	✓	lin. reg.	✓	72.1	73.0

Table 1: Evaluating the influence of different components of our approach on the OxUvA *dev* and GOT-10k *val* sets. See Section 4.3 for details.

As these heuristics can obscure the improvement of the underlying matching approach, we show results using *no temporal information* in Sec. 4.3 and 4.4, and show state-of-the-art results without heuristics on OxUvA. For completeness, we implement one simple heuristic which we ablate in Sec. 4.3. At every frame t , our detector outputs a set of candidate detections $D_t = \{d_{1,t}, \dots, d_{k,t}\}$, along with a confidence score $c_{i,t}$ for each detection. In our standard implementation, we select the detection d_t^* with the highest confidence. To incorporate temporal smoothness, we implement a simple heuristic: for each candidate box $d_{i,t}$, compute the mask intersection-over-union $j_{i,t}$ with the predicted detection d_{t-1}^* at the previous frame, and update the confidence as $c_{i,t} \leftarrow \alpha c_{i,t} + (1 - \alpha)j_{i,t}$. Then, we select d_t^* as the detection that maximizes this reweighted confidence. We set $\alpha = 0.6$ for all of our experiments. In order to avoid latching onto distractors, we temporarily disable this smoothness component if the track is broken, i.e. the IoU between the object locations at time t and $t+1$ is small ($< \alpha_{\text{low}}$); we re-enable the smoothness component if we maintain a smooth track for n frames, i.e. a track with consecutive object locations that have $\text{IoU} > \alpha_{\text{recover}}$. We set $\alpha_{\text{low}} = 0.1$, $\alpha_{\text{recover}} = 0.3$, and $n = 30$. We always show results both with and without this component for clarity.

4.3. Ablation study

In this section we analyze the influence of different components of our approach on the final performance. We use the *dev* sets of OxUvA and GOT-10k datasets for analysis, due to their complexity and diversity. Note that OxUvA requires explicitly thresholding confidence scores in order to detect when an object is not present in the video. For ablation, we report the area under the ROC curve (i.e., TPR vs. FPR curve) to better understand the performance of ablated components across score thresholds. GOT-10k does not require setting such a threshold, so we use the standard Average Overlap metric described in Section 4.1.

We start with a baseline variant of our approach, which

is trained only on videos labeled for tracking and achieves 63.2 OxUvA AUC and 64.7 GOT AO. Next, we evaluate the importance of object priors by pretraining our model on COCO as a generic object detector. This variant, shown in row 2, results in an 1.7% improvement in OxUvA AUC and a 3.7% improvement in GOT AO. Next, we train our model for detection and tracking jointly. As expected, this multi-task training strategy provides a modest bump on both OxUvA and GOT, leading to a model that improves in tracking while additionally being able to perform single-image detection. These improvements confirm our intuition that object priors are critical for tracking, and that the universal nature of our model is indeed helpful in transferring information from object detection datasets.

As described in Sec. 3.4, our framework is flexible, admitting various strategies for computing a discriminative template, \mathcal{T}_d . We analyze a few strategies for computing this template. In particular we compare our proposed linear regression framework (denoted as $\mathcal{T}_d = \text{'lin. reg.'}$) to two simple baselines: a non-discriminative one, that simply averages several features vectors sampled from the ground truth box (denoted with $\mathcal{T}_d = \text{'mean pos'}$), and a discriminative one that uses the difference between the means of positive and negative samples as a template (denoted with $\mathcal{T}_d = \text{'mean diff'}$). Note that these can be seen as special cases of linear regression. First, we observe that all these variants increase the model’s performance, but the linear regression approach results in the largest improvement of 5.3% OxUvA AUC and 0.9% GOT AO. Second, the ‘mean diff’ baseline shows the worst performance, which is counterintuitive. We attribute this result to the fact that simply subtracting the mean of the negative examples from the template leads to unstable behavior during training. In contrast, our principled approach to computing the template simplifies optimization. Finally, we show that incorporating the temporal smoothness (Section 4.2) provides significant improvements, particularly for short-term tracking as in GOT.

Discussion. Performing ablations on two diverse tracking datasets allows understanding the impact of ablated components for different challenges. For example, the use of detection priors seems to be significantly more pronounced in GOT, which requires tracking a diverse set of objects, than for OxUvA. This is to be expected: while video datasets are large enough to learn priors for common objects, image-based datasets like COCO provide priors for more diverse categories. Meanwhile, our discriminative templates provide a significant improvement on OxUvA, but a more modest improvement on GOT. We attribute this to the fact that our discriminative template is able to avoid latching onto distractors when the object of interest disappears, a phenomenon that is far more common in the long-term OxUvA dataset than on GOT.

Approach	TPR	TNR	GM
LCT [29]	22.7	43.2	31.3
MDNet [30]	42.1	0	32.4
TLD [20]	14.1	94.9	36.6
SiamFC + R [6]	35.4	43.8	39.7
DaSiam [53]	40.0	84.2	58.0
SiamMask	50.4	88.7	66.9
SiamRPN++	63.6	79.9	71.3
Ours w/o temporal	63.2	79.1	70.8
Ours	65.5	78.2	71.6

Table 2: In the top half, we show the current reported state-of-the-art results on OxUvA, from [40]. As these methods perform poorly, we first run recent state-of-the-art trackers (DaSiam [53], SiamMask [43], and SiamRPN++ [23]). We show that our approach significantly improves over both prior state-of-the-art, as well as these recent works.

Approach	AO	SR
DaSiam [53]	46.0	54.3
SiamMask [43]	66.8	78.3
SiamRPN++ [23]	65.8	77.5
Ours w/o temporal	69.5	79.1
Ours	73.0	82.8

Table 3: Results on GOT-10k val set. Prior methods train only on the GOT-10k training set. For a fair comparison, we compare to DaSiam, SiamMask and SiamRPN++, which are also trained on external data. By leveraging objectness priors and detection mechanisms, our method significantly improves, likely due to the diversity of objects in GOT.

4.4. Comparison to the state-of-the-art

We now compare our full approach to state-of-the art methods in object tracking and video object segmentation.

OxUvA evaluation We begin by presenting comparisons on the *dev* set of the OxUvA long term tracking benchmark in Table 2. We compare to the state-of-the-art approaches reported in [40]. As the approaches reported in [40] perform poorly qualitatively and quantitatively, we further evaluate two more recent trackers: DaSiam [53] and SiamMask [43] on this dataset. We use their publicly available code.

As shown in Table 2, our evaluation of DaSiam [53], SiamMask [43], and SiamRPN++ [23] out-performs the methods reported in [40]. Next, we evaluate our approach without using *any* temporal information in the "w/o temporal" row. This variant is completely stateless, and individually performs matching on each frame of the video. By contrast, almost all prior tracking approaches use heuristic tem-

Approach	P	R	F
SiamMask [43]	64.5	46.8	54.3
DaSiam-LT [53]	62.7	58.8	60.7
MBMD [52]	63.4	58.9	61.0
SiamRPN++ [23]	65.0	61.0	62.9
Ours w/o temporal	61.0	56.9	58.9
Ours	61.2	61.2	61.2

Table 4: F-measure on LTB35 (VOT 2018-LT challenge). Unlike many prior methods, we use a *single* model across all our experiments that is competitive on VOT while outperforming on other datasets.

poral smoothing to improve the performance of their models. Despite this lack of temporal information, our approach outperforms all but the very recent SiamRPN++ approach, including the recent work of [53, 43]. By adding the simple temporal heuristic described in Sec. 4.2, we further improve our results by 0.8%, achieving state-of-the-art results. We report results on the held out *test* set in supplementary.

GOT evaluation To validate our conclusions above, we further evaluate our approach on GOT-10k. As prior methods evaluated on GOT-10k use only the GOT-10k training set for training, we can not fairly compare our approach to them. Instead, we use [53, 43, 23] as baselines, which are the best method prior to ours on OxUvA. We report results on the validation set in Table 3, and additionally show results on the test set in the supplementary material. As can be seen from the table, we outperform all of these works by over 4%, which we attribute to the ability of our method to generalize to diverse object categories.

LTB-35 Evaluation We compare to state-of-the-art methods on the LTB-35 benchmark, which was used to evaluate long term tracking in the VOT 2018-LT challenge. This dataset focuses on tracking in videos over 2 minutes long on average, where the object of interest can frequently disappear and reappear in the video. We compare to state-of-the-art results in Table 4, as well as SiamMask[43]. Note that prior approaches, other than [43], provide dataset-specific hyperparameters that are tuned for this dataset. By contrast, we use a single model, a single set of hyperparameters, and a simple temporal heuristic across all datasets. Despite this, our method obtains a competitive F-measure of 61.2 on this dataset while outperforming on other datasets.

Mask evaluation on DAVIS’17 Finally, we evaluate our unified approach on the task of video object segmentation. To this end we use the validation set of DAVIS’17, and compare to the state-of-the-art approaches, including the ones that require finetuning the model on the test sequences. The

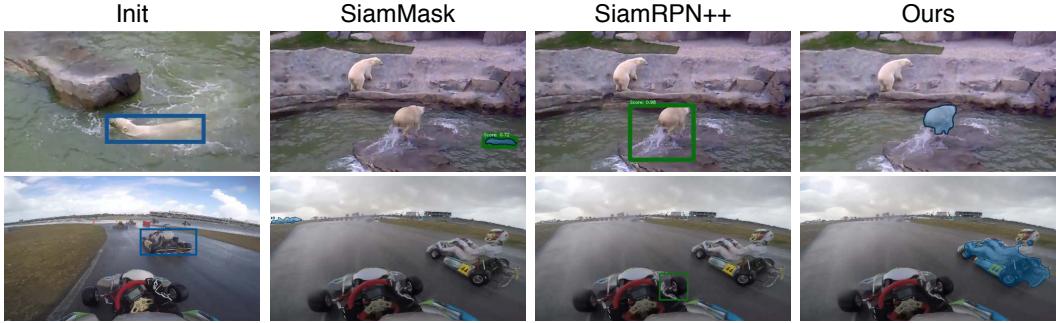


Figure 4: Qualitative comparison of our method with prior work. Top row: While SiamMask suffers from drift, and SiamRPN++ provides a loose bounding box around the object, our method provides a tight segmentation mask. Bottom row (failure case): All trackers fail to track the car of interest, which is nearly invisible in the distance. Intriguingly, while prior methods drift to background regions or parts (as with SiamMask and SiamRPN++), our method almost invariably latches onto *objects*.

Measure	PReMVOS [28]	CINM [4]	FeelVOS [41]	SiamMask [43]	Ours
Mask sup?	✓	✓	✓	✗	✗
Deep FT?	✓	✓	✗	✗	✗
Mean	73.9	67.2	69.1	54.3	59.2
\mathcal{J}	73.1	74.5	79.1	62.8	68.6
Decay	16.2	24.6	17.5	19.3	8.4
Mean	81.8	74.0	74.0	58.5	67.8
\mathcal{F}	88.9	81.6	83.8	67.5	76.1
Recall	19.5	26.2	20.1	20.9	12.0

Table 5: DAVIS ’17 validation results with intersection-over-union (\mathcal{J}) and F-measure (\mathcal{F}). Most prior methods require a labeled mask in the first frame (‘Mask sup’) or perform computationally expensive end-to-end fine-tuning per video (‘Deep FT’ row), our method efficiently and accurately segments objects without mask supervision.

results are presented in Table 5. We show qualitative results of our method in the supplementary material.

All methods in Table 5, with the exception of SiamMask and our method, require pixel-perfect segmentation in the first video frame and operate at a speed of less than 2 frames-per-second. By contrast, our method adds only a small overhead to the underlying detection model used. For our experiments, we used a ResNet-50 FPN backbone for Mask R-CNN, which led to a speed of approximately 7FPS. Despite using less supervision and computational time, our approach is competitive with dedicated video segmentation methods that use pixel-level masks in the first frame.

Detection evaluation on COCO As discussed in Section 3.3, our model can be used as a detector at test time. Although our focus is on tracking, we find that our model outputs high quality detections, providing a COCO instance segmentation mAP of 30.5, compared to 34.4 for an equivalent standalone detector that cannot track objects.

4.4.1 Qualitative results.

We show qualitative results in Fig. 4, comparing our results with SiamMask [43] and SiamRPN++ [23]. By leveraging objectness information, our method is able to successfully localize the full extent of objects, while providing precise *instance segmentation* masks for the object of interest. Further, we note an intriguing phenomenon due to our use of objectness: in failure cases, as in the bottom row of Figure 4, our method behaves qualitatively differently from prior work. When the object of interest is not visible, our method will report a mask with low confidence. Even in these cases, our method almost invariably segments *objects*, rather than drifting onto background regions or parts, as with prior methods.

5. Conclusion

This paper introduces a novel generic object tracking approach built on top of a state-of-the-art object-detection framework. The resulting model can be trained jointly for the two tasks, effectively incorporating objectness priors into tracking. Additionally, we propose learning discriminative templates in a fully differentiable manner that encode information both about the object of interest and about the distractors, increasing the tracker’s robustness. Finally, we extend our method to the related task of video object segmentation by simply adding a mask prediction branch.

Our resulting framework for tracking and video segmentation demonstrates state-of-the-art results on two recent tracking datasets (OxUvA and GOT10k), and also shows competitive performance on the DAVIS’17 benchmark for video object segmentation. We empirically show that these improvements are largely due to the generic objectness prior learned from the COCO dataset.

References

- [1] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2189–2202, 2012. 1
- [2] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *CVPR*, 2008. 1
- [3] B. Babenko, M.-H. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *CVPR*, 2009. 2
- [4] L. Bao, B. Wu, and W. Liu. Cnn in mrf: Video object segmentation via inference in a cnn-based higher-order spatio-temporal mrf. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5977–5986, 2018. 8
- [5] P. Bergmann, T. Meinhardt, and L. Leal-Taixe. Tracking without bells and whistles. *arXiv preprint arXiv:1903.05625*, 2019. 1
- [6] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr. Fully-convolutional Siamese networks for object tracking. In *ECCV*, 2016. 1, 2, 7
- [7] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual object tracking using adaptive correlation filters. In *CVPR*, 2010. 1, 2
- [8] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah. Signature verification using a “siamese” time delay neural network. In *Advances in neural information processing systems*, pages 737–744, 1994. 2
- [9] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. One-shot video object segmentation. In *CVPR*, 2017. 2
- [10] J. Cheng, Y.-H. Tsai, W.-C. Hung, S. Wang, and M.-H. Yang. Fast and accurate online video object segmentation via tracking parts. In *CVPR*, 2018. 2
- [11] M. Danelljan, G. Bhat, F. Shahbaz Khan, and M. Felsberg. Eco: Efficient convolution operators for tracking. In *CVPR*, 2017. 2
- [12] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010. 3
- [13] R. Girshick. Fast R-CNN. In *ICCV*, 2015. 3
- [14] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 3
- [15] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *ICCV*, 2017. 2, 3, 4, 5
- [16] D. Held, S. Thrun, and S. Savarese. Learning to track at 100 FPS with deep regression networks. In *ECCV*, 2016. 2
- [17] Y. Hua, K. Alahari, and C. Schmid. Online object tracking with proposal selection. In *ICCV*, 2015. 2
- [18] L. Huang, X. Zhao, and K. Huang. GOT-10k: A large high-diversity benchmark for generic object tracking in the wild. *arXiv preprint arXiv:1810.11981*, 2018. 2, 5
- [19] S. D. Jain and K. Grauman. Click carving: Segmenting objects in video with point clicks. In *Fourth AAAI Conference on Human Computation and Crowdsourcing*, 2016. 1
- [20] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1409–1422, 2012. 2, 7
- [21] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pfugfelder, L. C. Zajc, T. Vojir, G. Bhat, A. Lukezic, A. Eldesokey, G. Fernandez, and et al. The sixth visual object tracking vot2018 challenge results, 2018. 2, 5
- [22] M. Kristan, J. Matas, A. Leonardis, T. Vojir, R. Pfugfelder, G. Fernandez, G. Nebehay, F. Porikli, and L. Čehovin. A novel performance evaluation methodology for single-target trackers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11):2137–2155, Nov 2016. 1
- [23] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4282–4291, 2019. 2, 7, 8
- [24] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu. High performance visual tracking with Siamese region proposal network. In *CVPR*, 2018. 2
- [25] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 3
- [26] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 5
- [27] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In *ECCV*, 2016. 3
- [28] J. Luiten, P. Voigtlaender, and B. Leibe. Premvos: Proposal-generation, refinement and merging for video object segmentation. In *Asian Conference on Computer Vision*, pages 565–580. Springer, 2018. 8
- [29] C. Ma, X. Yang, C. Zhang, and M.-H. Yang. Long-term correlation tracking. In *CVPR*, 2015. 7
- [30] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, 2016. 7
- [31] F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung. Learning video object segmentation from static images. In *CVPR*, 2017. 2
- [32] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool. The 2017 DAVIS challenge on video object segmentation. *arXiv:1704.00675*, 2017. 2, 5
- [33] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 3
- [34] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 2, 3
- [35] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 5
- [36] J. S. Supancic and D. Ramanan. Self-paced learning for long-term tracking. In *CVPR*, 2013. 2

- [37] R. Tao, E. Gavves, and A. W. Smeulders. Siamese instance search for tracking. In *CVPR*, 2016. [2](#)
- [38] Y.-H. Tsai, M.-H. Yang, and M. J. Black. Video segmentation via object flow. In *CVPR*, 2016. [2](#)
- [39] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. Torr. End-to-end representation learning for correlation filter based tracking. In *CVPR*, 2017. [2](#)
- [40] J. Valmadre, L. Bertinetto, J. F. Henriques, R. Tao, A. Vedaldi, A. W. Smeulders, P. H. Torr, and E. Gavves. Long-term tracking in the wild: a benchmark. In *ECCV*, 2018. [2, 5, 7](#)
- [41] P. Voigtlaender, Y. Chai, F. Schroff, H. Adam, B. Leibe, and L.-C. Chen. Feelvos: Fast end-to-end embedding learning for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9481–9490, 2019. [2, 8](#)
- [42] P. Voigtlaender and B. Leibe. Online adaptation of convolutional neural networks for video object segmentation. In *BMVC*, 2017. [2](#)
- [43] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. Torr. Fast online object tracking and segmentation: A unifying approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1328–1338, 2019. [2, 3, 7, 8](#)
- [44] P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Learning to track for spatio-temporal action localization. In *ICCV*, 2015. [1](#)
- [45] L. Wen, D. Du, Z. Lei, S. Z. Li, and M.-H. Yang. Jots: Joint online tracking and segmentation. In *CVPR*, 2015. [2](#)
- [46] N. Wojke, A. Bewley, and D. Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649. IEEE, 2017. [1](#)
- [47] Y. Wu, J. Lim, and M.-H. Yang. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1834–1848, 2015. [1](#)
- [48] Y. Xiang, A. Alahi, and S. Savarese. Learning to track: Online multi-object tracking by decision making. In *Proceedings of the IEEE international conference on computer vision*, pages 4705–4713, 2015. [1](#)
- [49] N. Xu, L. Yang, Y. Fan, D. Yue, Y. Liang, J. Yang, and T. Huang. YouTube-VOS: A large-scale video object segmentation benchmark. In *ECCV*, 2018. [5](#)
- [50] L. Yang, Y. Wang, X. Xiong, J. Yang, and A. K. Katsaggelos. Efficient video object segmentation via network modulation. In *CVPR*, 2018. [2](#)
- [51] Y. Zhang, P. Tokmakov, M. Hebert, and C. Schmid. A structured model for action detection. *CVPR*, 2019. [1](#)
- [52] Y. Zhang, D. Wang, L. Wang, J. Qi, and H. Lu. Learning regression and verification networks for long-term visual tracking. *arXiv preprint arXiv:1809.04320*, 2018. [7](#)
- [53] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu. Distractor-aware siamese networks for visual object tracking. In *ECCV*, 2018. [1, 2, 7](#)