MAST: A Memory-Augmented Self-Supervised Tracker

Zihang Lai Erika Lu Weidi Xie
Visual Geometry Group, Department of Engineering Science
University of Oxford
{zlai, erika, weidi}@robots.ox.ac.uk

Abstract

Recent interest in self-supervised dense tracking has yielded rapid progress, but performance still remains far from supervised methods. We propose a dense tracking model trained on videos ***without any annotations*** *that surpasses previous self-supervised methods on existing benchmarks by a significant margin (+15%), and achieves performance comparable to supervised methods. In this paper, we first reassess the traditional choices used for self-supervised training and reconstruction loss by conducting thorough experiments that finally elucidate the optimal choices. Second, we further improve on existing methods by augmenting our architecture with a crucial memory component. Third, we benchmark on large-scale semi-supervised video object segmentation (*aka. *dense tracking), and propose a new metric: generalizability. Our first two contributions yield a self-supervised network that for the* ***first time*** *is competitive with supervised methods on standard evaluation metrics of dense tracking. When measuring generalizability, we show self-supervised approaches are actually* ***superior*** *to the majority of supervised methods. We believe this new generalizability metric can better capture the real-world use-cases for dense tracking, and will spur new interest in this research direction. Code will be released at* *https://github.com/zlai0/MAST*.

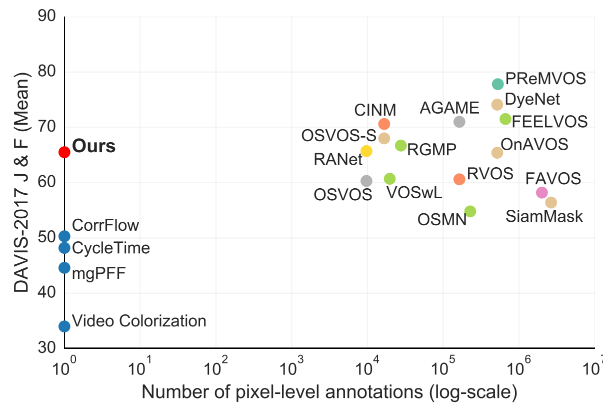


Figure 1: Comparison with other recent works on the DAVIS-2017 benchmarks, *i.e.* dense tracking or semi-supervised video segmentation given the first frame annotation. The proposed approach significantly outperforms other self-supervised approaches, and is even comparable to those trained with heavy supervision on ImageNet, COCO, Pascal, DAVIS, Youtube-VOS. In the x-axis, we only count pixel-wise segmentation.
Notation: CINM [3], OSVOS [6], FAVOS [8], AGAME [28], VOSwL [31], mgPFF [33], CorrFlow [37], DyeNet [39], PReMVOS [41]. OSVOS-S [42], RGMP [44], RVOS [54], FEELVOS [56], OnAVOS [57], Video Colorization [59], SiamMask [61], CycleTime [64], RANet [65], OSMN [73],

1. Introduction

Although the working mechanisms of the human visual system remain somewhat obscure at the level of neurophysiology, it is a consensus that tracking objects is a fundamental ability that a baby starts developing at two to three months of age [5, 34, 58]. Similarly, in computer vision systems, tracking plays key roles in many applications ranging from autonomous driving to video surveillance.

Given arbitrary objects defined in the first frame, a tracking algorithm aims to relocate the same object throughout the entire video sequence. In the literature, tracking can be cast into two categories: the first is Visual Object Tracking (VOT) [35], where the goal is to relocalize objects with bounding boxes throughout the video; the other aims for more fine-grained tracking, *i.e.* relocalize the objects with pixel-level segmentation masks, also known as Semi-supervised Video Object Segmentation (Semi-VOS) [48]. In this paper, we focus on the latter case, and will refer to it interchangeably with *dense tracking* from here on.

In order to train such dense tracking systems, most recent approaches rely on supervised training with extensive human annotations (see Figure 1). For instance, an ImageNet [10] pre-trained ResNet [18] is typically adopted as a feature encoder, and further fine-tuned on images or video frames annotated with fine-grained, pixelwise segmentation masks, *e.g.* COCO [40], Pascal [13], DAVIS [48] and YouTube-VOS [71]. Despite their success, this top-down training scheme seems counter-intuitive when considering the development of the human visual system, as infants can track and follow slow-moving objects before they are able to map objects to semantic meanings. With this evidence, it is unlikely the case that humans develop their tracking abil-

Figure 2: **Train once, test on multiple datasets:** Qualitative results from our *self-supervised dense tracking model* on DAVIS-2017 and YouTube-VOS dataset. The number on the top left refers to the frame number in the video. For all examples, the mask of the $0th$ frame is given, and the task is to track the objects along with the video. Our self-supervised tracking model is able to deal with challenging scenarios, such as large camera motion, occlusion and disocclusion, large deformation and scale variation.

ity in a top-down manner (supervised by semantics), at least not at the early-stage development of the visual system.

In contrast to the aforementioned approaches based on heavy supervision, self-supervised methods [37, 59, 60, 64] have recently been introduced, leading to more neurophysiologically intuitive directions. While not requiring any labeled data, the performance of these methods is still far from that of supervised methods (Figure 1).

We continue in the vein of self-supervised methods and propose an improved tracker, which we call *Memory-Augmented Self-Supervised Tracker* (MAST). Similar to previous self-supervised methods, our model performs tracking by learning a feature representation that enables robust pixel-wise correspondences between frames; it then propagates a given segmentation mask to subsequent frames based on the correspondences. We make three main contributions: *first*, we reassess the traditional choices used for self-supervised training and reconstruction loss by conducting thorough experiments to finally determine the optimal choices. *Second*, to resolve the challenge of tracker drift (*i.e.* as the object changes appearance or becomes occluded, each subsequent prediction becomes less accurate if propagated only from recent frames), we further improve on existing methods by augmenting our architecture with a crucial memory component. We design a coarse-to-fine approach that is necessary to efficiently access the memory bank: a two-step attention mechanism first coarsely searches for candidate windows, and then computes fine-grained matching. We conduct experiments to analyze our choice of memory frames, showing that both short- *and* long-term memory are crucial for good performance. *Third*, we benchmark on large-scale video segmentation datasets and propose a new metric, *i.e.* generalizability, with the goal of measuring the performance gap between tracking seen and unseen categories, which we believe better captures the real-world use-cases for category-agnostic tracking.

The result of the first two contributions is a self-supervised network that surpasses all existing approaches by a significant margin on DAVIS-2017 (15%) and YouTube-VOS (17%) benchmarks, making it competitive with supervised methods *for the first time*. Our results show that a strong representation for tracking can be learned without using any semantic annotations, echoing the early-stage development of the human visual system. Beyond significantly narrowing the gap with supervised methods on the existing metrics, we also demonstrate the *superiority* of self-supervised approaches over supervised methods on generalizability. On the unseen categories of YouTube-VOS benchmark, we surpass PreMVOS [41], the 2018 challenge winner algorithm trained on massive segmentation datasets. Furthermore, when we analyze the drop in performance between seen and unseen categories, we show that our method (along with other self-supervised methods) has a significantly smaller *generalization gap* than supervised methods. These results show that contrary to the popular belief that self-supervised methods are not yet useful due to their weaker performance, their greater generalization capability (due to not being at risk of overfitting to labels) is actually a more desirable quality when being deployed in real-world settings, where the domain gap can be significant.

2. Related Work

Dense tracking (*aka.* semi-supervised video segmentation) has typically been approached in one of two ways: propagation-based or detection/segmentation-based. The former approaches formulate the dense tracking task as a mask propagation problem from the first frame to the consecutive frames. To leverage the temporal consistency between two adjacent frames, many propagation-based methods often try to establish dense correspondences with optical flow or metric learning [20, 21, 29, 41, 56]. However, computing optical flow remains a challenging, yet unsolved problem. Our method relaxes the constraint of optical flow's one-to-one brightness constancy constraint and spatial smoothness, allowing each query pixel to potentially build correspondence with multiple reference pixels. On the other hand, detection/segmentation-based approaches address the tracking task with sophisticated detection or segmentation networks, but since these models are usually not class-agnostic during training, they often have to be fine-tuned on the first frame of the target video during inference [6, 41, 42], whereas our method requires no fine-tuning.

Self-supervised learning on videos has generated fruitful research in recent years. Due to the abundance of online data [1, 4, 11, 14, 15, 22, 24, 25, 26, 27, 32, 38, 43, 59, 63, 67, 68], various ideas have been explored to learn representations by exploiting the spatio-temporal information in videos. [14, 43, 66] exploit spatio-temporal ordering for learning video representations. Recently, Han *et al.* [17] learn strong video representations for action recognition by self-supervised contrastive learning on raw videos. Of more relevance, [37, 59] have recently leveraged the natural temporal coherency of color in videos, to train a network for tracking and correspondence related tasks. We discuss these works in more detail in Section 3.1. In this work, we propose to augment the self-supervised tracking algorithms with a differentiable memory module. We also rectify some flaws in their training process.

Memory-augmented models refer to the computational architecture that has access to a memory repository for prediction. Such models typically involve an internal memory implicitly updated in a recurrent process, *e.g.* LSTM [19] and GRU [9], or an explicit memory that can be read or written with an attention-based procedure [2, 12, 16, 36, 51, 53, 62, 70]. Memory models have been used for many applications, including reading comprehension [51], summarization [50], tracking[69], video understanding[7], and image and video captioning [70, 74]. In dense visual tracking, the popular memory-augmented models treat key frames as memory [45], and use attention mechanisms to read from the memory.

3. Method

The proposed dense tracking system, MAST (Memory-Augmented Self-Supervised Tracker), is a conceptually simple model for dense tracking that can be trained with self-supervised learning, *i.e.* **zero manual annotation** is required during training, and an object mask is only required for the first frame during inference. In Section 3.1, we provide relevant background of previous self-supervised dense tracking algorithms, and terminologies that will be used in later sections. Next, in Section 3.2, we pinpoint weaknesses in these works and propose improvements to the training signals. Finally, in Section 3.3, we propose memory augmentation as an extension to existing self-supervised trackers.

3.1. Background

In this section, we review previous papers that are closely related to this work [37, 59]. In general, the goal of self-supervised tracking is to learn feature representations that enable robust correspondence matching. During training, a proxy task is posed as reconstructing a target frame (I_t) by linearly combining pixels from a reference frame (I_{t-1}), with the weights measuring the strength of correspondence between pixels.

Specifically, a triplet ($\{Q_t, K_t, V_t\}$) exists for each input frame I_t, referring to *Query*, *Key*, and *Value*. In order to reconstruct a pixel i in the t-th frame ($\hat{I}_t^i$), an *Attention* mechanism is used for *copying* pixels from a subset of previous frames in the original sequence. This procedure is formalized as:

$$\hat{I}_t^i = \sum_j A_t^{ij} V_{t-1}^j \tag{1}$$

$$A_t^{ij} = \frac{exp\langle\, Q_t^i, K_{t-1}^j \rangle}{\sum_p exp\langle\, Q_t^i, K_{t-1}^p \rangle} \tag{2}$$

where $\langle\cdot,\cdot\rangle$ refers to the dot product between two vectors, query (Q) and key (K) are feature representations computed by passing the target frame I_t to a Siamese ConvNet $\Phi(\cdot;\theta)$, *i.e.* $Q_t = K_t = \Phi(I_t;\theta)$, A_t is the affinity matrix representing the feature similarity between pixel I_t^i and I_{t-1}^j, value (V) is the raw reference frame (I_{t-1}) during the training stage, and instance segmentation mask during inference, achieving reconstruction or dense tracking respectively.

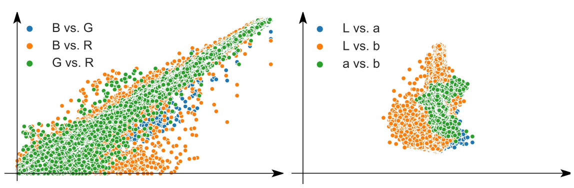
A key element in self-supervised learning is to set the proper *information bottleneck*, or the choice of what input information to withhold for learning the desired feature representation and avoiding trivial solutions. For example, in the reconstruction-by-copying task, an obvious shortcut is that the pixel in I_t can learn to match any pixel in I_{t-1} with the *exact same color*, yet not necessarily correspond to the same object. To circumvent such learning shortcuts, Vondrick *et al.* [59] intentionally drop the color information from the input frames. Lai and Xie [37] further show that a simple channel dropout can be more effective.

3.2. Improved Reconstruction Objective

In this section, we reassess the choices made in previous self-supervised dense tracking works and provide intuition for our optimal choices, which we empirically support in Section 5.

3.2.1 Decorrelated Color Space

Extensive experiments in the human visual system have shown that colors can be seen as combinations of the primary colors, namely red (R), green (G) and blue (B). For this reason, most of the cameras and emissive color displays represent pixels as a triplet of intensities: $(R, G, B) \in \mathcal{R}^3$. However, a disadvantage of the RGB representation is that the channels tend to be extremely correlated [49], as shown in Figure 3. In this case, the channel dropout proposed in [37] is unlikely to behave as an effective information bottleneck, since the dropped channel can almost always be determined by one of the remaining channels.



(a) RGB scatter plot (b) Lab scatter plot

Figure 3: Correlation between channels of RGB and Lab colorspace. We randomly take $100,000$ pixels from 65 frames in a sequence (snowboard) in the DAVIS dataset and plot the relative relationships between RGB channels. This phenomena generally holds for all natural images [49], due to the fact that all of the channels include a representation of brightness. Values are normalized for visualization purposes.

To remedy this limitation, we hypothesize that dropout in the decorrelated representations (*e.g*. Lab) would force the model to learn invariances suitable for self-supervised dense tracking; *i.e*. if the model cannot predict the missing channel from the observed channels, it is forced to learn a more robust representation rather than relying on local color information.

3.2.2 Classification vs. Regression

In the recent literature on colorization and generative models [46, 75], colors were quantized into discrete classes and treated as a multinomial distribution, since generating images or predicting colors from grayscale images is usually a non-deterministic problem; *e.g*. the color of a car can reasonably be red or white. However, this convention is suboptimal for self-supervised learning of correspondences, as we are not trying to generate colors for each pixel, but rather, estimate a precise relocation of pixels in the reference frames. More importantly, quantizing the colors leads to an information loss that can be crucial for learning high-quality correspondences.

We conjecture that directly optimizing a regression loss between the reconstructed frame ($\hat{I}_t$) and real frame (I_t) will provide more discriminative training signals. In this work, the objective $\mathcal{L}$ is defined as the Huber Loss:

$$\mathcal{L} = \frac{1}{n} \sum_i z_i \tag{3}$$

where

$$z_i = \begin{cases} 0.5(\hat{\mathbf{I}}_t^i - \mathbf{I}_t^i)^2, & \text{if } |\hat{\mathbf{I}}_t^i - \mathbf{I}_t^i| < 1 \\ |\hat{\mathbf{I}}_t^i - \mathbf{I}_t^i| - 0.5, & \text{otherwise} \end{cases} \tag{4}$$

where $\hat{\mathbf{I}}_t^i \in \mathcal{R}^3$ refers to RGB or Lab, normalized to the range [-1,1] in the reconstructed frame that is copied from pixels in the reference frame I_{t-1}, and I_t is the real frame at time point t.

3.3. Memory-Augmented Tracking

So far we have discussed the straightforward attention-based mechanism for propagating a mask from a single previous frame. However, as predictions are made recursively, errors caused by object occlusion and disocclusion tend to accumulate and eventually degrade the subsequent predictions. To resolve this issue, we propose an attention-based tracker that efficiently makes use of *multiple* reference frames.

3.3.1 Multi-frame tracker

An overview of our tracking model is shown in Figure 4. To summarize the tracking process: given the present frame and multiple past frames (memory bank) as input, we first compute the query (Q) for the present frame and keys (K) for all frames in memory. Here, we follow the general procedure in previous works as described in Section 3.1, where K and Q are computed from a shared-weight feature extractor and V is equal to the input frame (during training) or object mask (during testing). The computed affinity between Q and all the keys (K) in memory is then used to make a prediction for each query pixel depending on V. Note we don't put any weights on the reference frames, as this should be encoded in the affinity matrix (*e.g*. when a target and reference frame are dis-similar, the corresponding similarity value will be naturally low; thus the reference label will have less contribution to the labeling of a target pixel).

The decision of which pixels to include in K is crucial for good performance. Including all pixels previously seen is far too computationally expensive due to the quadratic explosion of the affinity matrix (*e.g*. the network of [37] produces affinity matrices with more than 1 billion elements for 480p videos). To reduce computation, [37] exploit temporal smoothness in videos and apply restricted attention, only computing the affinity with pixels in a ROI around the query pixel location. However, the temporal smoothness assumption holds only for temporally close frames.

To efficiently process temporally distant frames, we propose a two-step attention mechanism. The first stage involves coarse pixel-matching with the frames in the memory bank to determine which ROIs are likely to contain good

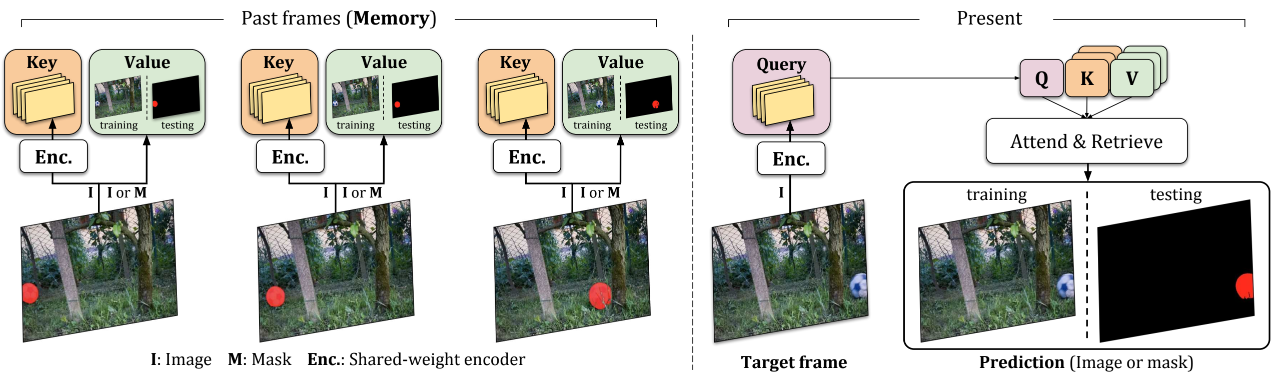


Figure 4: Structure of MAST. The current frame is used to compute **query** to attend and retrieve from memory (**key** & **value**). During training, we use raw video frame as **value** for self-supervision. Once the encoder is trained, we use instance mask as **value**. See Section 3.3 for details.

matches with the query pixel. In the second stage, we extract the ROIs and compute fine-grained pixel matching, as described in Section 3.1. Overall, the process can be summarized in Algorithm 1.

Algorithm 1 MAST

1: Choose m reference frames $Q_1, Q_2, ...Q_m$
2: Localize ROI $R_1, R_2, ...R_m$ according to 3.3.2 (Eq. 5 and 6) for each of the reference frames
3: Compute similarity matrix $A_t^{ij} = \langle Q^j, R_t^i \rangle$ between target frame Q and each ROI.
4: Output: pixel's label is determined by aggregating the labels of the ROI pixels (weighted by its affinity score).

3.3.2 ROI Localization

The goal of ROI localization is to estimate the candidate windows non-locally from memory banks. Intuitively, for short-term memory (temporally close frames), dilation is not required since spatial-temporal coherence naturally exists in videos; thus ROI localization becomes restricted attention (similar to [37]). However, for long-term memory, we aim to account for the fact that objects can potentially appear anywhere in the reference frames. We unify both scenarios into a single framework for learning ROI localization.

Formally, for the query pixel i in I_t, to localize the ROI from frame (I_{t-N}), we first compute in parallel $H^i_{t-N,x,y}$, the similarity heatmap between i and all candidate pixels in the dilated window:

$$H^i_{t-N,x,y} = softmax(Q^i_t \cdot im2col(K^i_{t-N}, \gamma_{t-N})) \quad (5)$$

where γ_{t-N} refers to the dilation rate for window sampling in frame I_{t-N}, and $im2col$ refers to an operation that transforms the input feature map into a matrix based on dilation rate. Specifically, in our experiments, the dilation rate is proportional to the temporal distance between the present frame and the past frames in the memory bank, *i.e.* $\gamma_{t-N} \propto N$. We use $\gamma_{t-N} = \lceil (t-N)/15 \rceil$.

The center coordinates for ROIs can be then computed via a *soft-argmax* operation:

$$P^i_{x,y} = \sum_{x,y} H^i_{x,y} * C \quad (6)$$

where $P^i_{x,y}$ is the estimated center location of the candidate window in frame I_{t-N} for query pixel I^i_t, and C refers to the grid coordinates (x, y) corresponding to the pixels in the window from $im2col$. The resampled Key ($\hat{K}^i_{t-N}$) for pixel I^i_t can be extracted with a bilinear sampler [23]. With all the candidate Keys dynamically sampled from different reference frames of the memory bank, we compute fine-grained matching scores only with these localized Keys, resembling a restricted attention in a non-local manner. With the proposed design, the model can therefore efficiently access high-resolution information for correspondence matching, without incurring large physical memory costs.

4. Implementation Details

Training: For fair comparison, we adopt as our feature encoder the same architecture (ResNet18) as [37] in all experiments (as shown in Supplementary Material). The network produces feature embeddings with a spatial resolution $1/4$ of the original image. The model is trained in a completely self-supervised manner, meaning the model is initialized with random weights, and we do *not* use any information other than raw video sequences. We report main results on two training datasets: OxUvA [52] and YouTube-VOS (both raw videos only). We report the first for fair comparison with the state-of-the-art method [37] and the second for maximum performance. As pre-processing, we resize all frames to $256 \times 256 \times 3$. In all of our experiments, we use I_0, I_5 (only if the index for the current frame is larger than 5) as long term memory, and I_{t-5}, I_{t-3}, I_{t-1} as short term memory. Empirically, we find the choice of frame number has small impact on performance, but using both long and short term memory is essential.

During training, we first pretrain the network with a pair of input frames, *i.e.* one reference frame and one target frame are fed as inputs. One of the color channels is randomly dropped with probability $p = 0.5$. We train our model end-to-end using a batch size of 24 for 1M iterations with the Adam optimizer. The initial learning rate is set to $1e$-3, and halved after 0.4M, 0.6M and 0.8M iterations. We then finetune the model using multiple reference frames (our full memory-augmented model) with a small learning rate of $2e$-5 for another 1M iterations. As discussed in Section 3.2.2, the model is trained with a photometric loss between the reconstruction and the true frame.

Inference: We use the trained feature encoder to compute the affinity matrix between pixels in the target frame and those in the reference frames. The affinity matrix is then used to propagate the desired pixel-level entities, such as instance masks in the dense tracking case (Algorithm 1).

Image Feature Alignment: Due to memory constraints, the supervision signals in previous methods were all defined on bilinearly downsampled images. This introduces a misalignment between strided convolution layers and images from naïve bilinear downsampling. We handle this spatial misalignment between feature embedding and image by directly sampling at the strided convolution centers. This seemingly minor change actually brings significant improvement to the downstream tracking task (Table 4). More implementation details can be found in arXiv version (https://arxiv.org/abs/2002.07793).

5. Experiments

We benchmark our model on two public benchmarks: DAVIS-2017 [48] and the current largest video segmentation dataset, YouTube-VOS [71]. The former contains 150 HD videos with over 30K manual instance segmentations, and the latter has over 4000 HD videos of 90 semantic categories, totalling over 190k instance segmentations. For both datasets, we benchmark the proposed self-supervised learning architecture (MAST) on the official semi-supervised video segmentation setting (*aka.* dense tracking), where a ground truth instance segmentation mask is given for the first frame, and the objective is to propagate the mask to subsequent frames. In Section 5.1, we report performance of our full model and several ablated models on the DAVIS benchmark. Next, in Section 5.2, we analyze the *generalizability* of our model by benchmarking on the large-scale YouTube-VOS dataset.

Standard evaluation metrics. We use region similarity ($\mathcal{J}$) and contour accuracy ($\mathcal{F}$) to evaluate the tracked instance masks [47].

Generalizability metrics. To demonstrate the generalizability of tracking algorithms in category-agnostic scenarios, *i.e.* the categories in training set and testing set are disjoint, YouTube-VOS also explicitly benchmarks the performances on unseen categories. We therefore evaluate a *generalization gap* in Section 5.3, which is defined as the average performance difference between seen and unseen object classes:

$$\text{Gen. Gap} = \frac{(\mathcal{J}_{seen} - \mathcal{J}_{unseen}) + (\mathcal{F}_{seen} - \mathcal{F}_{unseen})}{2}$$

Note, the proposed metric aims to explicitly penalize the case where the performance on seen outperforms unseen by large margins, while at the same time provide a reward when the performance on unseen categories is higher than on seen ones.

5.1. Video Segmentation on DAVIS-2017

5.1.1 Main results

In Table 1, we compare MAST with previous approaches on the DAVIS-2017 benchmark. Two phenomena can be observed: *first*, our proposed model clearly outperforms all other self-supervised methods, surpassing previous state-of-the-art CorrFlow by a significant margin (65.5 vs 50.3 on $\mathcal{J}\&\mathcal{F}$). *Second*, despite using only ResNet18 as the feature encoder, our model trained with self-supervised learning can still surpass supervised approaches that use heavier architectures.

5.1.2 Ablation Studies

To examine the effects of different components, we conduct a series of ablation studies by removing one component at a time. All models are trained on OxUvA (except for the analysis on different datasets), and evaluated on DAVIS-2017 semi-supervised video segmentation (*aka.* dense tracking) *without* any finetuning.

Choice of color spaces. As shown in Table 2, we perform different experiments with input frames transformed into different color spaces, *e.g.* RGB, Lab or HSV. We find that the MAST model trained with Lab color space always outperforms the other color spaces, validating our conjecture that dropout in a decorrelated color space leads to better feature representations for self-supervised dense tracking, as explained in Section 3.2.1. Additionally, we compare our default setting with a model trained with cross-color space matching task (shown in Table 3). That means to use a different color space for the input and the training objective, *e.g.* input frames are in RGB, and loss function is defined in Lab color space. Interestingly, the performance drops significantly, we hypothesis this can attribute to the fact that all RGB channels include a representation of brightness, making it highly correlate to the luminance in Lab, therefore acting as a weak information bottleneck.

Loss functions. As a variation of our training procedure, we experiment with different loss functions: cross entropy loss on the quantized colors, and photometric loss with Huber loss. As shown in Table 2, regression with real-valued photometric loss surpasses classification significantly,

Method	Backbone	Supervised	Dataset (Size)	$\mathcal{J}\&\mathcal{F}$(Mean) ↑	$\mathcal{J}$(Mean) ↑	$\mathcal{J}$(Recall) ↑	$\mathcal{F}$(Mean) ↑	$\mathcal{F}$(Recall) ↑
Vid. Color. [59]	ResNet-18	✗	Kinetics (800 hours)	34.0	34.6	34.1	32.7	26.8
CycleTime† [64]	ResNet-50	✗	VLOG (344 hours)	48.7	46.4	50.0	50.0	48.0
CorrFlow† [37]	ResNet-18	✗	OxUvA (14 hours)	50.3	48.4	53.2	52.2	56.0
UVC⋆ [72]	ResNet-18	✗	Kinetics (800 hours)	59.5	57.7	68.3	61.3	69.8
MAST (Ours)	ResNet-18	✗	OxUvA (14 hours)	**63.7**	**61.2**	**73.2**	**66.3**	**78.3**
MAST (Ours)	ResNet-18	✗	YT-VOS (5.58 hours)	**65.5**	**63.3**	**73.2**	**67.6**	**77.7**
ImageNet [18]	ResNet-50	✓	I (1.28M, 0)	49.7	50.3	-	49.0	-
OSMN [73]	VGG-16	✓	ICD (1.28M, 227k)	54.8	52.5	60.9	57.1	66.1
SiamMask [61]	ResNet-50	✓	IVCY (1.28M, 2.7M)	56.4	54.3	62.8	58.5	67.5
OSVOS [6]	VGG-16	✓	ID (1.28M, 10k)	60.3	56.6	63.8	63.9	73.8
OnAVOS [57]	ResNet-38	✓	ICPD (1.28M, 517k)	65.4	61.6	67.4	69.1	75.4
OSVOS-S [42]	VGG-16	✓	IPD (1.28M, 17k)	68.0	64.7	74.2	71.3	80.7
FEELVOS [56]	Xception-65	✓	ICDY (1.28M, 663k)	71.5	69.1	79.1	74.0	83.8
PReMVOS [41]	ResNet-101	✓	ICDPM (1.28M, 527k)	77.8	73.9	83.1	81.8	88.9
STM [45]	ResNet-50	✓	IDY (1.28M, 164k)	81.8	79.2	-	84.3	-

Table 1: Video segmentation results on DAVIS-2017 validation set. Dataset notations: I=ImageNet, V = ImageNet-VID, C=COCO, D=DAVIS, M=Mapillary, P=PASCAL-VOC Y=YouTube-VOS. For size of datasets, we report (length of *raw videos*) for self-supervised methods and (#image-level annotations, #pixel-level annotations) for supervised methods. ⋆ denotes concurrent work. † denotes highest results reported after original publication. Higher values are better.

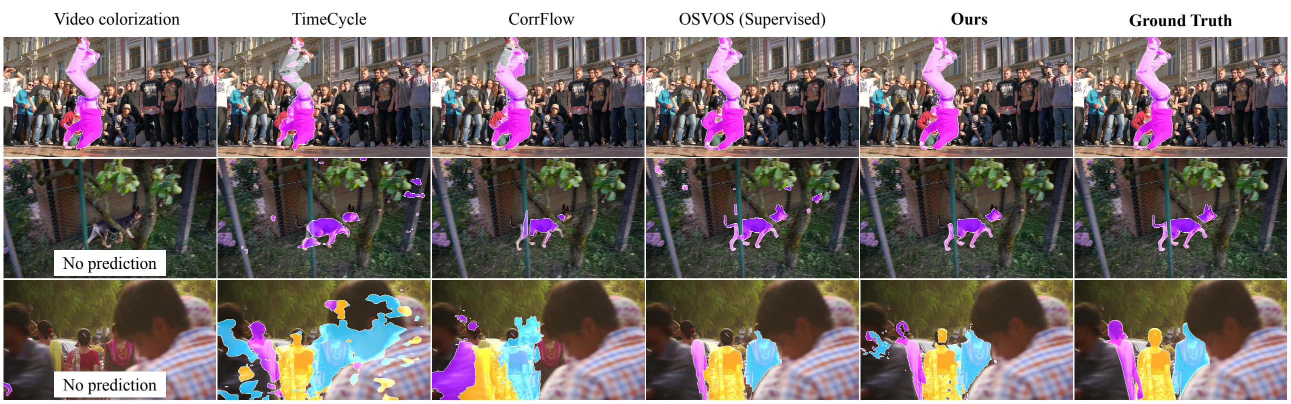


Figure 5: Our method vs. previous self-supervised methods. Other methods show systematic errors in handling occlusions. Row 1: The dancer undergoes large self-occlusion. Row 2: The dog is repeatedly occluded by poles. Row 3: Three women reappear after being occluded by the man in the foreground.

validating our conjecture that the information loss during color quantization results in inferior representations for self-supervised tracking (as explained in Section 3.2), due to less discriminative training signals.

Image feature alignment. To evaluate the alignment module proposed for aligning features with the original image, we compare it to direct bilinear image downsampling used by CorrFlow [37]. The result in Table 4 shows that our approach achieves about 2.2% higher performance.

Dynamic memory by exploiting more frames. We compare our default network with variants that have only short term memory or long term memory. Results are shown in Table 5. While both short term memory and long term memory alone can make reasonable predictions, the combined model achieves the highest performance. The qualitative predictions (Figures 2 and 5) also confirm that the improvements come from reduced tracker drift. For instance, when severe occlusion occurs, our model is able to attend and retrieve high-resolution information from frames that are temporally distant.

5.2. Youtube Video Object Segmentation

We also evaluate the MAST model on the Youtube-VOS validation split (474 videos with 91 object categories). As no other self-supervised methods have been tested on the benchmark, we directly compare our results with supervised methods. As shown in Table 8, our method outperforms the other self-supervised learning approaches by a significant margin (64.2 vs. 46.6), and even achieves comparable performance to many heavily supervised methods.

5.3. Generalizability

As another metric for evaluating category-agonostic tracking, the YouTube-VOS dataset conveniently has separate measures for seen and unseen object categories. We can therefore estimate testing performance on out-of-distribution samples to gauge the model's generalizability to more challenging, unseen, real-world scenarios. As seen from the last two columns, we rank second amongst all algorithms in unseen objects. In these unseen classes, we are even 3.9% higher than the DAVIS 2018 and YouTube-VOS

Colors.	Loss	$\mathcal{J}$(Mean)	$\mathcal{F}$(Mean)
RGB	Cls.	42.5	45.3
	Reg.	52.7	57.1
HSV	Cls.	32.5	35.3
	Reg.	54.3	58.6
Lab	Cls.	47.1	48.9
	Reg.	**61.2**	**66.3**

Table 2: **Training colorspaces and loss**: Our final model trained with Lab colorspace with regression loss outperforms all other models on dense tracking task. Higher values are better.

Input	Loss	$\mathcal{J}$(Mean)	$\mathcal{F}$(Mean)
Lab	RGB	48.2	52.0
RGB	Lab	46.8	49.9
Lab	Lab	**61.2**	**66.3**

Table 3: **Cross color space matching *vs.* single color space**: Cross color space matching shows inferior results compared to single color space.

I-F Align	$\mathcal{J}$(Mean)	$\mathcal{F}$(Mean)
No	59.1	64.0
Yes	**61.2**	**66.3**
	+2.1	+2.3

Table 4: **Image-Feature alignment**: Using the improved Image-Feature alignment implementation improves the results. Higher values are better.

Memory	$\mathcal{J}$(Mean)	$\mathcal{F}$(Mean)
Only long	44.6	48.7
Only short	57.3	61.8
Both	**61.2**	**66.3**

Table 5: **Memory length**: Removing either long term or short term memory results in a performance drop.

Propagation	$\mathcal{J}$(Mean)	$\mathcal{F}$(Mean)
Soft	57.0	61.7
Hard	**61.2**	**66.3**
	+4.2	+4.6

Table 6: **Soft *vs.* hard propagation**: *Quantizing* class probability of each pixel (hard propagation) shows large gains over propagating probility distribution (soft propagation).

Dataset	$\mathcal{J}$(Mean)	$\mathcal{F}$(Mean)
OxUvA	61.2	66.3
ImageNet VID	60.0	63.9
YouTube-VOS (w/o anno.)	63.3	67.6

Table 7: **Training dataset**: All datasets provide reasonable performance, with O and Y slightly superior. We conjecture that our model gains from higher quality videos and larger object classes in these datasets.

Method	Sup.	Overall ↑	Seen		Unseen		Gen. Gap ↓
			$\mathcal{J}$ ↑	$\mathcal{F}$ ↑	$\mathcal{J}$ ↑	$\mathcal{F}$ ↑	
Vid. Color.[59]†	✗	38.9	43.1	38.6	36.6	37.4	3.9
CorrFlow[37]	✗	46.6	50.6	46.6	43.8	45.6	3.9
MAST (Ours)	✗	**64.2**	**63.9**	**64.9**	**60.3**	**67.7**	**0.4**
OSMN[73]	✓	51.2	60.0	60.1	40.6	44.0	17.75
MSK[30]	✓	53.1	59.9	59.5	45.0	47.9	13.25
RGMP[44]	✓	53.8	59.5	-	45.2	-	14.3
OnAVOS[57]	✓	55.2	60.1	62.7	46.6	51.4	12.4
RVOS[55]	✓	56.8	63.6	67.2	45.5	51.0	17.15
OSVOS[6]	✓	58.8	59.8	60.5	54.2	60.7	2.7
S2S[71]	✓	64.4	71.0	70.0	55.5	61.2	12.15
PreMVOS[41]	✓	66.9	71.4	75.9	56.5	63.7	13.55
STM[45]	✓	79.4	79.7	84.2	72.8	80.9	5.1

Table 8: Video segmentation results on Youtube-VOS dataset. Higher values are better. According to the evaluation protocol of the benchmark, we report performance separated into "seen" and "unseen" classes ("Seen" with respect to training set). † indicates results based on our reimplementation. The first- and second-best results on the unseen category are highlighted in red and blue, respectively.

2018 video segmentation challenge winner, PreMVOS[41], a complex algorithm trained with multiple large manually labeled datasets. For fair comparison, we train our model only on the YouTube-VOS training set. We also re-train two most relevant self-supervised methods in the same manner as baselines. Even learning from only a subset of all classes, our model generalizes well to unseen classes, with a generalization gap (*i.e.* the performance difference between seen and unseen objects) near zero (0.4). This gap is much smaller than any of the baselines (avg $= 11.5$), suggesting a unique advantage to most other algorithms trained with labels.

By training on large amounts of unlabeled videos, we learn an effective tracking representation without the need for any human annotations. This means that the learned network is not limited to a specific set of object categories (*i.e.* those in the training set), but is more likely to be a "universal feature representation" for tracking. Indeed, the only supervised algorithm that is comparable to our method in generalizability is OSVOS (2.7 *vs.* 0.4). However, OSVOS uses the first image from the testing sequence to perform costly domain adaptation, *e.g.* one-shot fine-tuning. In contrast, our algorithm requires no fine-tuning, which further demonstrates its zero-shot generalization capability.

Note our model also has a smaller generalization gap compared to other self-supervised methods as well. This further attests to the robustness of its learned features, suggesting that our improved reconstruction objective is highly effective in capturing general features.

6. Conclusion

In summary, we present a memory-augmented self-supervised model that enables accurate and generalizable pixel-level tracking. The algorithm is trained without any semantic annotation, and surpasses previous self-supervised methods on existing benchmarks by a significant margin, narrowing the gap with supervised methods. On unseen object categories, our model actually outperforms all but one existing methods that are trained with heavy supervision. As computation power grows and more high quality videos become available, we believe that self-supervised learning algorithms can serve as a strong competitor to their supervised counterparts for their flexibility and generalizability.

References

[1] Pulkit Agrawal, Joao Carreira, and Jitendra Malik. Learning to see by moving. In *Proc. ICCV*, 2015. 3

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *Proc. ICLR*, 2015. 3
[3] Linchao Bao, Baoyuan Wu, and Wei Liu. Cnn in mrf: Video object segmentation via inference in a cnn-based higher-order spatio-temporal mrf. In *Proc. CVPR*, 2018. 1
[4] Bert De Brabandere, Xu Jia, Tinne Tuytelaars, and Luc Van Gool. Dynamic filter networks. In *NIPS*, 2016. 3
[5] T. Berry Brazelton, Mary Louise Scholl, and John S. Robey. Visual responses in the newborn. *Pediatrics*, 1966. 1
[6] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *Proc. CVPR*, 2017. 1, 3, 7, 8
[7] Wu Chao-Yuan, Feichtenhofer Christoph, Fan Haoqi, He Kaiming, Krähenbühl Philipp, and Girshick Ross. Long-Term Feature Banks for Detailed Video Understanding. In *Proc. CVPR*, 2019. 3
[8] Jingchun Cheng, Yi-Hsuan Tsai, Wei-Chih Hung, Shengjin Wang, and Ming-Hsuan Yang. Fast and accurate online video object segmentation via tracking parts. In *Proc. CVPR*, 2018. 1
[9] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 3
[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *Proc. CVPR*, 2009. 1
[11] Emily Denton and Vighnesh Birodkar. Unsupervised learning of disentangled representations from video. In *NIPS*, 2017. 3
[12] J. Devlin, M.W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 2018. 3
[13] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results. http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html, 2009. 1
[14] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *Proc. CVPR*, 2017. 3
[15] Chuang Gan, Boqing Gong, Kun Liu, Hao Su, and Leonidas J. Guibas. Geometry guided convolutional neural networks for self-supervised video representation learning. In *Proc. CVPR*, 2018. 3
[16] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014. 3
[17] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *1st International Workshop on Large-scale Holistic Video Understanding, ICCV*, 2019. 3
[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016. 1, 7
[19] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 3
[20] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G. Schwing. Maskrnn: Instance level video object segmentation. In *NIPS*, 2017. 3
[21] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G. Schwing. Videomatch: Matching based video object segmentation. In *Proc. ECCV*, 2018. 3
[22] Phillip Isola, Daniel Zoran, Dilip Krishnan, and Edward H. Adelson. Learning visual groups from co-occurrences in space and time. In *Proc. ICLR*, 2015. 3
[23] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *NIPS*, 2015. 5
[24] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Conditional image generation for learning the structure of visual objects. In *NIPS*, 2018. 3
[25] Dinesh Jayaraman and Kristen Grauman. Learning image representations tied to ego-motion. In *Proc. ICCV*, 2015. 3
[26] Dinesh Jayaraman and Kristen Grauman. Slow and steady feature analysis: higher order temporal coherence in video. In *Proc. CVPR*, 2016. 3
[27] Longlong Jing, Xiaodong Yang, Jingen Liu, and Yingli Tian. Self-supervised spatiotemporal feature learning by video geometric transformations. *arXiv preprint arXiv:1811.11387*, 2018. 3
[28] Joakim Johnander, Martin Danelljan, Emil Brissman, Fahad Shahbaz Khan, and Michael Felsberg. A generative appearance model for end-to-end video object segmentation. In *Proc. CVPR*, 2019. 1
[29] Anna Khoreva, Rodrigo Benenson, Eddy Ilg, Thomas Brox, and Bernt Schiele. Lucid data dreaming for multiple object tracking. In *arXiv preprint arXiv: 1703.09554*, 2017. 3
[30] Anna Khoreva, Federico Perazzi, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *arXiv preprint arXiv:1612.02646*, 2016. 8
[31] A. Khoreva, A. Rohrbach, and B. Schiele. Video object segmentation with language referring expressions. In *Proc. ACCV*, 2018. 1
[32] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *AAAI*, 2018. 3
[33] Shu Kong and Charless Fowlkes. Multigrid predictive filter flow for unsupervised learning on videos. *arXiv 1904.01693, 2019*, 2019. 1
[34] Janet P. Krementizer, Herbert G. Vaughan, Diane Kurtzberg, and Kathryn Dowling. Smooth-pursuit eye movements in the newborn infant. *Child Development*, 1979. 1
[35] Matej Kristan, Jiri Matas, Aleš Leonardis, Tomas Vojir, Roman Pflugfelder, Gustavo Fernandez, Georg Nebehay, Fatih Porikli, and Luka Čehovin. A novel performance evaluation methodology for single-target trackers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016. 1
[36] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. In *International conference on machine learning*, 2016. 3
[37] Zihang Lai and Weidi Xie. Self-supervised learning for video correspondence flow. In *Proc. BMVC.*, 2019. 1, 2, 3, 4, 5, 7, 8
[38] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *Proc. ICCV*, 2017. 3
[39] Xiaoxiao Li and Chen Change Loy. Video object segmen-

tation with joint re-identification and attention-aware mask propagation. In *Proc. ECCV*, 2018. 1
[40] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollar. Microsoft coco: Common objects in context. In *Proc. ECCV*, 2014. 1
[41] Jonathon Luiten, Paul Voigtlaender, and Bastian Leibe. Premvos: Proposal-generation, refinement and merging for video object segmentation. In *Proc. ACCV*, 2018. 1, 2, 3, 7, 8
[42] Kevis-Kokitsi Maninis, Sergi Caelles, Yuhua Chen, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. Video object segmentation without temporal information. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018. 1, 3, 7
[43] Ishan Misra, C. Lawrence Zitnick, and Martial Hebert. Shuffle and learn: Unsupervised learning using temporal order verification. In *Proc. ECCV*, 2016. 3
[44] Seoung Wug Oh, Joon-Young Lee, Kalyan Sunkavalli, and Seon Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *Proc. CVPR*, 2018. 1, 8
[45] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proc. ICCV*, 2019. 3, 7, 8
[46] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *Proc. ICML*, 2016. 4
[47] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proc. CVPR*, 2016. 6
[48] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017. 1, 6
[49] Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Computer graphics and applications*, 21(5):34–41, 2001. 4
[50] Abigail See, Peter J Liu, and Christopher D Manning. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*, 2017. 3
[51] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448, 2015. 3
[52] Jack Valmadre, Luca Bertinetto, Joao F. Henriques, Ran Tao, Andrea Vedaldi, Arnold Smeulders, Philip Torr, and Efstratios Gavves. Long-term tracking in the wild: A benchmark. In *Proc. ECCV*, 2018. 5
[53] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*, 2017. 3
[54] Carles Ventura, Miriam Bellver, Andreu Girbau, Amaia Salvador, Ferran Marques, and Xavier Giro-i Nieto. Rvos: End-to-end recurrent network for video object segmentation. In *Proc. CVPR*, 2019. 1
[55] Carles Ventura, Miriam Bellver, Andreu Girbau, Amaia Salvador, Ferran Marques, and Xavier Giro-i Nieto. Rvos: End-to-end recurrent network for video object segmentation. In *Proc. CVPR*, June 2019. 8
[56] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen. Feelvos: Fast end-to-end embedding learning for video object segmentation. In *Proc. CVPR*, 2019. 1, 3, 7
[57] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for video object segmentation. In *Proc. BMVC.*, 2017. 1, 7, 8
[58] C. von Hofsten. Eye–hand coordination in the newborn. *Developmental Psychology*, 1982. 1
[59] Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. Tracking emerges by colorizing videos. In *Proc. ECCV*, 2018. 1, 2, 3, 7, 8
[60] Ning Wang, Yibing Song, Chao Ma, Wengang Zhou, Wei Liu, and Houqiang Li. Unsupervised deep tracking. In *Proc. CVPR*, 2019. 2
[61] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip H.S. Torr. Fast online object tracking and segmentation: A unifying approach. In *Proc. CVPR*, 2019. 1, 7
[62] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proc. CVPR*, 2018. 3
[63] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *Proc. ICCV*, 2015. 3
[64] Xiaolong Wang, Allan Jabri, and Alexei A. Efros. Learning correspondence from the cycle-consistency of time. In *Proc. CVPR*, 2019. 1, 2, 7
[65] Ziqin Wang, Jun Xu, Li Liu, Fan Zhu, and Ling Shao. Ranet: Ranking attention network for fast video object segmentation. In *Proc. ICCV*, 2019. 1
[66] Donglai Wei, Joseph Lim, Andrew Zisserman, and William T. Freeman. Learning and using the arrow of time. In *Proc. CVPR*, 2018. 3
[67] Olivia Wiles, A. Sophia Koepke, and Andrew Zisserman. Self-supervised learning of a facial attribute embedding from video. In *Proc. BMVC.*, 2018. 3
[68] Olivia Wiles, A. Sophia Koepke, and Andrew Zisserman. X2face: A network for controlling face generation using images, audio, and pose codes. In *Proc. ECCV*, 2018. 3
[69] Zhu Xizhou, Wang Yujie, Dai Jifeng, Yuan Lu, and Wei Yichen. Flow-guided feature aggregation for video object detection. In *Proc. ICCV*, 2017. 3
[70] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proc. ICML*, 2015. 3
[71] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv:1809.03327*, 2018. 1, 6, 8
[72] Li Xueting, Liu Sifei, De Mello Shalini, Wang Xiaolong, Kautz Jan, and Yang Ming-Hsuan. Joint-task self-supervised learning for temporal correspondence. In *NeurIPS*, 2019. 7
[73] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos Katsaggelos. Efficient video object segmentation via network modulation. In *Proc. CVPR*, 2018. 1, 7, 8
[74] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. Describing videos by exploiting temporal structure. In *Proc. ICCV*, 2015. 3
[75] Richard Zhang, Phillip Isola, and Alexei Efros. Colorful image colorization. In *Proc. ECCV*, 2016. 4