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.

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. relocate 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-
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, 49, 63, 67, 68], various ideas have been explored to learn representations by exploiting the spatio-temporal information in videos. [4, 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 ($I_t$), an Attention mechanism is used for copying pixels from a subset of previous frames in the original sequence. This procedure is formalized as:

\[
\hat{I}_t = \sum_j A_{t}^{ij} V_{i-1}^{j} \tag{1}
\]

\[
A_{t}^{ij} = \frac{exp(Q^t_i, K_{t-1}^j)}{\sum_p exp(Q^t_i, K_{t-1}^p)} \tag{2}
\]

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: \((R, G, B) \in \mathbb{R}^3\). 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.

![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 DAIS 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.](Image)

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 \(L\) is defined as the Huber Loss:

\[
L = \frac{1}{n} \sum_i z_i
\]

where
\[
z_i = \begin{cases} 
0.5(|\hat{I}_i^t - I_i^t|^2), & \text{if } |\hat{I}_i^t - I_i^t| < 1 \\
|\hat{I}_i^t - I_i^t| - 0.5, & \text{otherwise}
\end{cases}
\]

where \(I_i^t \in \mathbb{R}^3\) 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
matches with the query pixel. In the second stage, we extract the ROIs and compute fine-grained pixel matching, as described in Section 3.3. 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 $A^{ij}_t = (Q^j_i, R^i_t)$ 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 $H_{t-N,x,y}^i$, the similarity heatmap between $i$ and all candidate pixels in the dilated window:

$$H_{t-N,x,y}^i = \text{softmax}(Q^i_t \cdot \text{im}2\text{col}(K_{t-N}^i, \gamma_{t-N}))$$  \(5\)

where $\gamma_{t-N}$ refers to the dilation rate for window sampling in frame $I_{t-N}$, and $\text{im}2\text{col}$ 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 $\gamma_{t-N} = [(t - N)/15]$.

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

$$P_{x,y}^i = \sum_{x,y} H_{x,y}^i \ast C$$  \(6\)

where $P_{x,y}^i$ is the estimated center location of the candidate window in frame $I_{t-N}$ for query pixel $I_t$, and $C$ refers to the grid coordinates $(x, y)$ corresponding to the pixels in the window from $im2col$. The resampled Key $(K_{t-N}^i)$ for pixel $I_t$ 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:

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

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,
results show that our approach achieves about \( \text{(Mean)} \) higher performance. To evaluate the alignment mod-
ule proposed for aligning features with the original image, we compare it to direct bilinear image downsampling used by CorrFlow \cite{corrflow}. 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-agnostic 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.
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 net-work 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


[39] Xiaoqiao Li and Chen Change Loy. Video object segment-


