Anchor Diffusion for Unsupervised Video Object Segmentation

Zhao Yang∗
University of Oxford
zhao.yang@eng.ox.ac.uk

Qiang Wang∗
CASIA
qiang.wang@nlpr.ia.ac.cn

Luca Bertinetto
Five AI
luca@robots.ox.ac.uk

Weiming Hu
CASIA
wmhu@nlpr.ia.ac.cn

Song Bai
University of Oxford
songbai.site@gmail.com

Philip H.S. Torr
University of Oxford
philip.torr@eng.ox.ac.uk

Abstract

Unsupervised video object segmentation has often been tackled by methods based on recurrent neural networks and optical flow. Despite their complexity, these kinds of approach tend to favour short-term temporal dependencies and are thus prone to accumulating inaccuracies, which cause drift over time. Moreover, simple (static) image segmentation models, alone, can perform competitively against these methods, which further suggests that the way temporal dependencies are modelled should be reconsidered. Motivated by these observations, in this paper we explore simple yet effective strategies to model long-term temporal dependencies. Inspired by the non-local operators of [63], we introduce a technique to establish dense correspondences between pixel embeddings of a reference “anchor” frame and the current one. This allows the learning of pairwise dependencies at arbitrarily long distances without conditioning on intermediate frames. Without online supervision, our approach can suppress the background and precisely segment the foreground object even in challenging scenarios, while maintaining consistent performance over time. With a mean IoU of 81.7%, our method ranks first on the DAVIS-2016 leaderboard of unsupervised methods, while still being competitive against state-of-the-art online semi-supervised approaches. We further evaluate our method on the FBMS dataset and the video saliency dataset ViSal, showing results competitive with the state of the art.

1. Introduction

Video object segmentation (VOS) is a fundamental task in many important areas such as autonomous driving [11, 37, 15], robotic manipulation [23], video surveillance [54] and video editing [43]. Contemporary literature typically considers this problem in either the semi-supervised or the unsupervised setting. In both cases, the objective is to predict in every frame pixel-level masks delineating certain objects of interest.

Under the semi-supervised setting, at test time methods can rely on a mask that specifies the object to segment. In contrast, the unsupervised setting does not provide any initialisation. Without online supervision, the task might be considered ambiguous, as different objects could be considered of interest for different reasons, according to the application. Among researchers, the current consensus is to segment foreground objects where a human gaze is more likely to focus [4]. In more practical terms, an object is generally considered as foreground if it is sufficiently large, in motion and centred in the scene. In certain datasets (e.g., FBMS [39] and ViSal [61]), in the same video, multiple foreground objects are considered, while in DAVIS-2016 [45] only a single object is considered.

With the aim of tracking temporal changes in target objects, current state-of-the-art unsupervised approaches generally model motion cues in a video sequence via optical flow [29, 28, 47, 22, 52, 51, 20, 9] or recurrent neural networks (RNNs) [16, 28, 52]. Typically, these methods sequentially propagate features from the previous steps to the current one, making the current prediction dependent on the entire history of the video. Though having the potential of exploiting informative temporal cues, these approaches suffer from several limitations. RNNs often rely on training techniques such as truncated backpropagation through time to reduce the cost of parameter updates, which limits their long-term modelling capability [49]. Moreover, while the gating mechanism in LSTMs alleviates the issue of vanishing gradients [1, 42], these can still have exploding gradients, which often requires clipping or rescaling the norm of the gradients during training [50]. Optical flow vectors only predict one-step motion cues at each frame in a video, which can accumulate errors over time. What is more, mod-
els relying on optical flow are typically trained on synthetic videos due to the high cost of per-frame and per-pixel labelling. Therefore, when applying these systems to real videos, the domain gap can cause the flow fields to contain several inaccuracies, especially when the foreground is nearly static [52].

In the video object segmentation community, the deterioration of performance over time in unsupervised VOS methods based on optical flow or RNNs is well known and has been widely discussed [28, 55, 40, 7]. For instance, Li et al. [28] demonstrate that as a regular optical flow-based model progresses through frames, foreground embeddings become increasingly closer in feature space to the first frame’s background as opposed to the foreground. Furthermore, Voigtlaender et al. [55] observed that a simple static segmentation model can achieve competitive results in the unsupervised VOS setting, which further corroborates the case for steering away from the sequential modelling strategies used by established methods.

Motivated by the above observations, in this work we opt for a much simpler solution, which is based on learning the similarity of pixels between frames that can be arbitrarily far apart. To ensure representation consistency and reduce long-term drift, we propagate the features of the first frame (the “anchor”) to the current frame via an aggregation technique inspired by the non-local operation introduced by Wang et al. [63]. This approach allows us to forgo of sequential modelling, while at the same time enabling us to deal with long-term dependencies and achieve high robustness over time, as shown in our experiments.

Despite its simplicity and online operability, our method, which we name anchor diffusion network (AD-Net), outperforms the current state of the art [47] on the DAVIS-2016 leaderboard by a margin of (absolute) 2.2% in terms of intersection-over-union, without resorting to auxiliary training data or post-processing. Moreover, it also achieves state-of-the-art results on FBMS [39] and the ViSal [61] video saliency benchmark. Code and pre-trained models are available at https://github.com/yz93/anchor-diff-VOS.

2. Related work

The problem of video object segmentation (VOS) is tackled by the computer vision community in the unsupervised or semi-supervised settings, which are defined by the level of supervision provided at test time.

Semi-supervised VOS methods are provided with a pixel-wise mask identifying the target object in the first frame of a video. When aiming at very high segmentation accuracy, methods [3, 38, 55, 44, 45, 21, 30] generally perform online fine-tuning on the basis of this supervision, sometimes exploiting data-augmentation techniques [3, 21] or self-supervision [55]. As online fine-tuning can take up to several minutes per video, many recently proposed methods renounce to it and instead aim at a faster online speed (e.g., [57, 8, 7]). These faster semi-supervised approaches come in many flavours. For instance, Chen et al. [7] learn a metric space for pixel embeddings, which is then used to establish associations between pixels across frames, while Cheng et al. [8] suggest to individually track object parts from the first frame with a visual object tracker [2] and then aggregate them according to their similarity with the initialisation mask.

Unsupervised VOS methods, instead, cannot rely on any supervision at test time and are often based on optical flow and RNNs. The purely optical flow-based MP-Net [51] discards appearance modelling and casts segmentation as foreground motion prediction, an approach which poorly deals with static foreground objects. To address this problem, several methods (e.g., LVO [52], SegFlow [9], MotAdapt [47] and MBN [29]) suggest to integrate appearance-based and optical flow-based features together, leading to variations of the “two-stream model” presenting two dedicated parallel branches. The drawbacks of such methods are threefold. First, flow estimation networks are typically trained on synthetic datasets and can thus result in poor performance when deployed in the real world. Second, while modelling long-term temporal dependencies is critical for adapting to significant online changes, the vector fields can only model short-term one-step de-
dependencies. Targeting this issue, Tokmakov et al. [52] proposed to extend the horizon spanned by optical flow-based features by employing a convolutional gated recurrent unit [10]. Third, vector fields cannot distinguish foreground and background objects when they move in a synchronised fashion (e.g., the cars in a traffic jam). Li et al. [29] attempt to address this issue by employing a bilateral network for detecting the motion of background objects. Our investigations with a much simpler appearance-based approach show that optical flow may not be an essential component of unsupervised VOS systems.

RNN-based models are often challenged by the problems of exploding and vanishing gradients [1, 42], which limit their long-term modelling capability. Among the methods that make use of recurrent connections, Song et al. [48] propose a novel convolutional long short-term memory [16] architecture, in which two atrous convolution [5] layers are stacked along the forward axis and propagate features in opposite directions.

Recently, it has been shown [28, 40, 55] that both recurrent and optical flow-based methods significantly suffer from a deterioration in the quality of their prediction over time. This has motivated the several approaches (including ours) that tackle video object segmentation by simply learning similarities between pixel embeddings (e.g., [13, 7, 28, 29]). These methods first select a set of seed pixels that are most likely to belong to the foreground object and then classify all other pixels based on their similarities to these seeds, for instance by thresholding or by propagating labels between neighbours. Fathi et al. [13] adopt this approach for semantic instance segmentation, in which the pairwise pixel similarity function measures the likelihood of two pixels belonging to the same instance. IET [28] extends this concept to video sequences. Similarly, it selects a set of foreground and background seeds for each frame and organises them into tracks. It then segments each frame individually based on pixel similarities with the foreground and background seeds. Note that IET utilises pre-trained instance embeddings. MBN [29] extends IET with a bilateral filtering network that filters false-positive foreground predictions using optical flow features and an energy minimisation procedure on a graph of seeds sampled from a few consecutive frames. When segmenting frame \( t \), MBN classifies each pixel by assigning it the label of the seed (sampled from frames \( t-1, t, \) and \( t+1 \)) with which it has the smallest embedding distance.

The main drawback of these methods is in the complexity involved in the procedures of seed selection, ranking and classification, critical for achieving good performance. Moreover, these algorithms also depend on multiple scores such as motion saliency and objectness that need to be carefully calibrated and combined into one final metric.

Albeit our proposal is related to this last class of approaches, is considerably simpler. Instead of separately learning individual components from image datasets and classifying pixels based on similarities with seeds, our method performs similarity learning, feature propagation and binary segmentation in a single classification network.

### 3. Method

We are interested in the task of binary segmentation of a sequence of video frames, where the final performance is measured by the average segmentation quality of individual frames. Therefore, our method should perform well under two aspects. First, similarly to what is expected from static segmentation models, it should be able to provide accurate segmentation masks of foreground objects in individual frames. Second, it should be able to well adapt to the changes in appearance of the foreground objects throughout the whole video.

In the proposed anchor diffusion network (AD-Net) (schematised in Figure 2), we address both requirements in a single end-to-end model by leveraging the recently proposed non-local operations [63]. Closely related to the concept of self-attention [53], a non-local operation is a neural network building block that captures the dependencies within a set of input feature vectors.

To achieve our first goal, a non-local operation is applied to the encoding of the target frame, in a similar way it is applied for semantic image segmentation [14], forming the intra-frame branch of our overall model. To achieve our second goal, we propagate information between two frames: a fixed anchor frame and the current frame, forming the anchor-diffusion branch of our overall model. We name the branch this way to give relevance to its functionality of “diffusing” information from the anchor to the large number of target frames at test time, which encourages foreground embeddings of each target frame to be consistent over time.

In the following, we describe our pipeline in more detail.

**Pipeline.** The input of our model consists of a pair of images: an anchor frame \( I_0 \), which is fixed for any specific video, and the frame to segment \( I_t \). The overall pipeline is schematically illustrated in Figure 2. First, a feature encoder (the fully-convolutional DeepLabv3 [5]) encodes \( I_0 \) and \( I_t \) into the corresponding embeddings \( X_0 \in \mathbb{R}^{hw \times c} \) and \( X_t \in \mathbb{R}^{hw \times c} \), where \( c \) denotes the number of channels and \( h, w \) denote the height and width of the frame. We refer to the \( c \)-dimensional feature vector at each location as a pixel embedding. The output of this first stage is then fed to three parallel branches: a skip connection with an identity mapping [18], the intra-frame branch, and the anchor-diffusion branch. \( X_t \) is fed to all branches, while \( X_0 \) only to the anchor-diffusion branch. Finally, the resulting features from the three branches are concatenated together along the channel dimension before the classification layer.
The entire network is trained end-to-end with a binary cross-entropy loss. Though any frame could be selected as the anchor frame, in practice we always choose the first frame for computational convenience and because, in benchmarks, the first frame is guaranteed to contain the foreground objects. During training, the first frame and a random frame are sampled from the video.

### 3.1. Anchor diffusion

As described earlier, \( X_0 \) and \( X_t \) represent the embeddings of the anchor and the current frame respectively. In order to reinforce the foreground signal, it is important to know which pixel embeddings in \( X_t \) correspond to the background introduced throughout a video. To achieve this, in the anchor-diffusion branch we compute a transition matrix \( P \in \mathbb{R}^{hw \times hw} \) which establishes dense correspondences between each pair of pixels from \( X_0 \) and \( X_t \) and use it to map \( X_t \) to a new encoding \( \tilde{X}_t \), in which the pixel embeddings are weighted according to their similarity with the foreground:

\[
\tilde{X}_t = PX_t. \tag{1}
\]

As qualitatively illustrated in Figure 1 and in the supplementary material, this procedure significantly strengthens the foreground while weakening the background. It is worth noting that one can also simply use the concatenation of \( X_0 \) and \( X_t \) to achieve this goal. However, we find in our experiments that the correspondence learning in Equation (1) can better localise the foreground objects.

Similarly to [63], the transition matrix is defined as

\[
P = \text{softmax}(\frac{1}{2} X_0 X_t^T), \tag{2}
\]

where \( X_0 X_t^T \) is a pairwise dot product similarity between each pair of pixel embeddings in \( X_0 \) and \( X_t \). Following [53, 31], we scale the dot product with a factor \( z = \sqrt{c} \), where \( c \) is the number of channels of \( X_0 \) and \( X_t \). The rationale being that, for embeddings with high dimensionality, dot products can be very large and thus push the output of the softmax to regions where gradients are small [53]. The softmax function normalises each row of \( \frac{1}{2} X_0 X_t^T \) to sum to one, thereby preserving scale invariance of the pixel embeddings. Without normalisation, multiplying \( \frac{1}{2} X_0 X_t^T \) with \( X_t \) can entirely change the scale of the pixel embeddings.

In the case of the intra-frame branch, each output pixel embedding can be considered as a global aggregation of all input pixel embeddings weighted by pairwise appearance similarity. It has been shown that such use of non-local operations [63] can harness long-range spatial information, which is beneficial for semantic segmentation [14]. Empirically, as detailed in the ablation studies of Table 1, we found that incorporating this branch in addition to the anchor-diffusion branch further improves the performance of the model.

The intra-frame branch improves segmentation accuracy, but does not address the temporal changes in a video sequence. Conversely, the anchor-diffusion branch models pairwise dependencies between frames, with the result of enhancing the consistency of pixel embeddings and reducing drift.

**Qualitative analysis.** As shown in Figure 1, as desirable, the foreground pixel (green) results in high similarity scores in correspondence to the anchor pixels belonging to the foreground object and low everywhere else, which highlights the temporal consistency of the foreground representation over time. Conversely, the background car (red) and asphalt (purple) exhibit dispersed similarity correspondences over different regions in the anchor. As these responses span large areas that contain objects of different appearance and semantic classes, an aggregation of features from these locations average out discriminative information
from the background that may lead to false-positive predictions, effectively suppressing signals from the original background pixel embeddings.

In Figure 3, instead, we report how foreground embeddings change over time by computing the average cosine distance between the foreground embeddings of a later frame and those of the first frame. The embeddings of our proposal are significantly stabler, while those of the baseline quickly grow apart. This suggests that AD-Net is capable of preserving foreground information in the first frame over long periods of time in a video.

4. Experiments

In the following, after discussing important implementation details regarding our architecture and training procedure, in Section 4.1 we illustrate the three benchmarks we adopted, in Section 4.2 we describe several ablation studies and finally, in Section 4.3, we provide an extensive comparison with the state of the art.

Implementation details. We employ the fully-convolutional DeepLabv3 [5] as the feature encoder, and initialise its ResNet101 [18] backbone with weights pre-trained on ImageNet. The other layers in DeepLabv3 are randomly initialised. The configuration of the dilation rates follows the original model [5] and presents a total stride of 8. We modify the number of output channels in the last layer to 128, which corresponds to $c$ in Section 3.

In the anchor-diffusion step, the spatial dimensions of each image encoding are flattened and transposed where appropriate in order to perform batched matrix multiplication. The outputs of the three branches are concatenated along the channel axis and reduced to dimension 128 via a $1 \times 1$ convolution with LeakyReLU non-linearity and a dropout with rate 0.1. The final classification layer is implemented as a $1 \times 1$ convolution with output channel 1 followed by a sigmoid layer.

Training. Each training example consists of a pair of images: We randomly sample a video and use the first frame and a randomly sampled frame (not including the first) from this video as the training example. We experimented with randomly sampling both frames instead of fixing the first and observed slightly worse performance. Each input frame is cropped to a randomly sized region which encloses the ground-truth foreground. Random rotations are performed at 45-degree increments, with a probability of 51% not rotating and equal probabilities rotating to any of the remaining angles.

The model is trained with binary cross-entropy loss. Network parameters are optimised via stochastic gradient descent with weight decay 0.0005. The initial learning rate is set to 0.005 and follows a “poly” adjustment policy [5], where the initial learning rate is multiplied by $(1 - \frac{\text{iter}}{\text{iter}_{\text{total}}})^{0.9}$ at each iteration. The model is trained for 30,000 iterations and the batch size is 8. Raw predictions are upsampled via bilinear interpolation to the size of the ground-truth masks.

Inference. At test time, the features of the anchor frame are computed once and reused throughout the video. Multi-scale and mirrored inputs are employed to enhance the final performance. Each input image is scaled by factors of 0.75, 1.00 and 1.50 and horizontally flipped. The final heatmap is the mean of all output heatmaps. Thresholding at 0.5 produces the final binary labels.

Instance pruning. Since semantic segmentation approaches like the one we use lack the notion of instance, and some videos from the DAVIS-2016 dataset [45] present multiple objects that can be deemed as foreground (a rather ill-defined scenario for UVOS), we experiment with a simple set of post-processing steps that prune non-foreground objects. As instance trajectories measure the spatial changes of an instance, they can be used to detect background instances which have distinct trajectory patterns than the foreground instance. First, we establish online temporal correspondences by using a pre-trained object detection model [66] to predict the locations of all objects and track the trajectory of each detection across the entire video using an intersection-over-union criterion between consecutive bounding boxes. Once object tracks have been established, we use the cumulative area of instance masks across frames as a proxy to identify foreground objects, thus pruning small objects or objects that are only present in a fraction of the video. This process produces a filtering mask, which is multiplied element-wise with AD-Net predictions to obtain the final predictions. More details and hyper-parameters related to this process (which we refer to as instance pruning) are provided in the supplementary material.

4.1. Benchmarks

Datasets. DAVIS [45] is a benchmark and yearly challenge for video object segmentation. Unsupervised methods are trained and evaluated with the DAVIS-2016 dataset, which annotates a single foreground entity. There are 30 videos for training (2,079 training frames) and 20 videos for validation. We train our method on the training set and evaluate on the validation set. The FBMS [39] dataset is another
challenging benchmark for unsupervised video object segmentation. It contains 29 training videos and 30 test videos with annotations on 720 frames. Following conventions in [47, 48, 52, 28, 29], we evaluate on the test set. The ViSal [61] dataset is a video salient object detection benchmark. It contains 17 video sequences with annotations on 193 frames. We report saliency evaluations of our method on ViSal for demonstrating the robustness and wide applicability of our method.

**Evaluation metrics.** For DAVIS, we adopt the official evaluation metrics of mean region similarity $\mathcal{J}$, which is the intersection-over-union of the prediction and ground truth, and mean contour accuracy $\mathcal{F}$, which is the F-measure defined on contour points in the prediction and ground truth. To provide more insights and for fair comparisons, we plot precision-recall (PR) curves on all three benchmark datasets with the corresponding F-measure. On the FBMS dataset, the main evaluation metric is the F-measure. On the ViSal dataset, we report the mean absolute error (MAE) and the F-measure. For definitions of MAE and the F-measure, we refer interested readers to [19].

### 4.2. Ablation studies

We conduct several ablations to precisely evaluate the effectiveness of the anchor-diffusion procedure. First, we evaluate DeepLabv3 [6] as-is, simply fine-tuning it on the DAVIS training set. This semantic segmentation baseline (designed for static images) performs on par with some state-of-the-art unsupervised VOS methods (see Table 2). This is in line with what described by Voigtlaender et al. [55], but it is rather curious that it still applies after two years of progress. Clearly, the competitive performance can be partially attributed to the high performance of DeepLabv3 for the similar task of semantic segmentation of static images. However, this result also shows that existing unsupervised VOS techniques are not able to successfully model and leverage temporal dependencies and that different approaches should be sought.

Starting from this baseline, we evaluate four variants that differ in the embeddings they consider at the terminal concatenation layer (see Figure 2). Each corresponds to a row below Baseline in Table 1. The first variant (“intra-frame”) computes non-local features within the same frame $X_t$ and without the anchor-diffusion branch. The second (“anchor”) simply concatenates $X_0$ to $X_t$. The third performs anchor diffusion on $X_0$ and $X_t$, and concatenates with $X_t$, without features from the intra-frame branch. The fourth (our final model, AD-Net) concatenates both the output of the intra-frame branch and that of the anchor-diffusion branch with $X_t$.

The “intra-frame” variant improves over the baseline, which shows the potential of utilising context information within the current frame. The “anchor” variant demonstrates the general usefulness of an anchor frame, despite the apparent limitation that the fixed representation of the anchor frame does not adapt to changes in the current frame. The solid performance gains validate our motivation to further develop the anchor-diffusion mechanism. The “anchor-diffusion” variant illustrates the efficacy of the proposed feature diffusion mechanism across the anchor and current frames. It brings a performance boost of 2.02 (absolute) points over the baseline, larger than the contribution brought by the “intra-frame” and “anchor” variants.

### 4.3. Comparison with the state of the art

In Table 2, we evaluate AD-Net against state-of-the-art unsupervised VOS methods on the DAVIS public leaderboard and also provide the performance of several popular semi-supervised methods as a term of reference. AD-Net attains the highest performance among all unsupervised methods on the DAVIS validation set, while also performing very competitively on the FBMS test set. In particular, on DAVIS we outperform the second-best method (MotAdapt [47]) by an absolute margin of 2.2% in $\mathcal{J}$ and 0.8% in $\mathcal{F}$ before applying the post-processing step of instance pruning. After applying instance pruning as described earlier, AD-Net achieves the final performance of 81.7 in $\mathcal{J}$.
and 80.5 in $F$, leading the second-best method by 4.5 and 3.1 absolute points respectively. In addition, despite being unsupervised at inference time, AD-Net outperforms many semi-supervised methods which instead require to be initialised with a mask in the first frame.

After our proposed AD-Net, the second and third-best ranking methods are MotAdapt [47] and PDB [48], which are particularly representative of two classes of methods.

PDB is representative of top-performing RNN-based methods. Although, in theory, RNNs could model long-range time dependencies, in practice they are constrained to model relatively short sequences. First, as the computational graph of an (unrolled) RNN grows in depth with the length of a video sequence, backpropagation is typically limited to a few time steps (e.g., 5 in RGMP [40]). Such backpropagation cannot guarantee long-term dependency modelling [49]. Second, despite the gating and memory mechanisms adopted by LSTMs and GRUs, long propagation paths of gradients still cause exploding or vanishing gradients [1, 42].

Conversely, MotAdapt is representative of top-performing methods that employ optical flow. It consists of a two-stream architecture, which dedicates two network branches (trained jointly but with different parameters) to process RGB images and pre-computed optical flow fields. The two-branch network is further fine-tuned at inference time, with pseudo-labels generated by a teacher network. Although optical flow is an intuitive way to model inter-frame dependencies and aid segmentation, results in Tables 1 and 2 demonstrate that simply developing a better appearance-based model can overshadow the benefits of a dedicated optical flow branch. Moreover, the strategy of fine-tuning at inference time adopted by MotAdapt and many semi-supervised methods is a time-consuming process, taking many seconds up to minutes per video. In contrast, AD-Net leverages a simpler architecture, which makes it fast at inference time. Without instance pruning, it runs online and at 4 frames per second on an NVIDIA TITAN X GPU, with frames at the original DA VIS resolution of $854 \times 480$. Clearly, speed can be easily traded off at a small cost in performance, by using lower resolution and/or lighter baseline architectures.

The precision-recall analysis of AD-Net is presented in Figure 4, where we demonstrate that our approach generally outperforms also existing salient object detection methods. AD-Net achieves superior performance in all regions of the PR curve on the DA VIS validation set, maintaining significantly higher precision at all recall thresholds. On the challenging FBMS test set, AD-Net maintains a clear advantage below the 90% recall threshold. On the ViSal dataset, it is noteworthy that nearly perfect precision is maintained up until the 60% recall rate, which is higher than the other methods.

### Table 3. Salient object detection performance of AD-Net, compared against 18 popular saliency prediction methods.

<table>
<thead>
<tr>
<th>Methods</th>
<th>DA VIS $F \uparrow$</th>
<th>DA VIS $MAE \downarrow$</th>
<th>FBMS $F \uparrow$</th>
<th>FBMS $MAE \downarrow$</th>
<th>ViSal $F \uparrow$</th>
<th>ViSal $MAE \downarrow$</th>
</tr>
</thead>
<tbody>
<tr>
<td>FGRNE</td>
<td>0.043</td>
<td>78.6</td>
<td>0.083</td>
<td>77.9</td>
<td>0.040</td>
<td>85.0</td>
</tr>
<tr>
<td>FCNS</td>
<td>0.053</td>
<td>72.9</td>
<td>0.100</td>
<td>73.5</td>
<td>0.041</td>
<td>87.7</td>
</tr>
<tr>
<td>SGP</td>
<td>0.126</td>
<td>67.7</td>
<td>0.171</td>
<td>57.1</td>
<td>0.172</td>
<td>64.8</td>
</tr>
<tr>
<td>GAFL</td>
<td>0.091</td>
<td>57.8</td>
<td>0.150</td>
<td>55.1</td>
<td>0.099</td>
<td>72.6</td>
</tr>
<tr>
<td>SAGE</td>
<td>0.105</td>
<td>47.9</td>
<td>0.142</td>
<td>58.1</td>
<td>0.096</td>
<td>73.4</td>
</tr>
<tr>
<td>STUW</td>
<td>0.098</td>
<td>69.2</td>
<td>0.143</td>
<td>52.8</td>
<td>0.132</td>
<td>67.1</td>
</tr>
<tr>
<td>SP</td>
<td>0.130</td>
<td>60.1</td>
<td>0.161</td>
<td>53.8</td>
<td>0.126</td>
<td>73.1</td>
</tr>
<tr>
<td>AD-Net (ours)</td>
<td>0.044</td>
<td>80.8</td>
<td>0.064</td>
<td>81.2</td>
<td>0.030</td>
<td>90.4</td>
</tr>
</tbody>
</table>

**Evaluation as video saliency.** The definition of salient objects in a video for benchmarks like ViSal [61] is very related to the one of “foreground objects” for benchmarks like DA VIS or FBMS (see Section 1). Annotations in salient object detection datasets can vary from coarse annotations such as bounding boxes to fine-grained pixel-level real-valued scores, and sometimes even take the form of human eye fixations. ViSal provides pixel-level annotations as binary labels, annotating large, moving objects as the foreground and everything else as the background. Despite the many types of annotations, evaluation metrics are fairly standard and use pixel-level annotations either in binarised form (PR curve and F-measure) or as normalised saliency scores between 0 and 1 (MAE), which are directly applicable to the scores produced by AD-Net.

As shown in Table 3, the proposed AD-Net improves the state of the art for both DA VIS and FBMS also for standard saliency scores, showing consistency with Table 2. The largest improvements lie in FBMS, where both MAE and F-measure significantly outperform previous records. On DA VIS, F-measure is the highest among all methods with a significant leading margin. On the ViSal dataset, AD-Net achieves best MAE (lower is better) among all video saliency models and obtains F-measure close to the overall best method. Remarkably, despite not having trained for the task of saliency prediction, we outperform previous saliency methods under saliency metrics on DA VIS and FBMS, and achieve very competitive results on ViSal.

### 5. Conclusion

In this paper, we proposed Anchor Diffusion Network (AD-Net), a method for unsupervised video object segmentation based on non-local operations. Instead of modelling
temporal dependencies with recurrent connections or adopting pre-computed optical flow like contemporary work, we argue for a significantly simpler and more effective approach, which consists in establishing correspondences of pixel embeddings between a reference frame and the current one. With this strategy, we can easily model long-term temporal dependencies at a low computational cost. We show how, during inference, this procedure is able to suppress the background while preserving the foreground even when abrupt changes in appearance occur. Quantitative evaluations across three standard benchmarks demonstrate the advantage of our proposed method on the task of unsupervised video object segmentation with respect to the state of the art. Moreover, our method is also surprisingly competitive against the state of the art in semi-supervised video object segmentation and video saliency.

**Acknowledgements.** This work was supported by the ERC grant ERC-2012-AdG 321162-HELIOS, EPSRC grant Seebibyte EP/M013774/1, EPSRC/MURI grant EP/N019474/1, and Tencent. We would also like to acknowledge the Royal Academy of Engineering and Five AI.
**References**


[28] Siyang Li, Bryan Seybold, Alexey Vorobyov, Alireza Fathi, Qin Huang, and C-C Jay Kuo. Instance embedding transfer to unsupervised video object segmentation. In *CVPR*, 2018. 1, 2, 3, 6


[33] Zhi Liu, Junhao Li, Linwei Ye, Guangling Sun, and Li-juan Shen. Saliency detection for unconstrained videos using superpixel-level graph and spatiotemporal propagation. *TCSVT*, 2017. 7

[34] Zhi Liu, Xiang Zhang, Shuhua Luo, and Olivier Le Meur. Superpixel-based spatiotemporal saliency detection. *TCSVT*, 2014. 7
[37] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. IJRR, 2017. 1
[47] Mennatullah Siam, Chen Jiang, Steven Lu, Laura Petrich, Mahmoud Gamal, Mohamed Elhoseiny, and Martin Jagersand. Video object segmentation using teacher-student adaptation in a human robot interaction (hri) setting. In ICRA, 2019. 1, 2, 6, 7
[52] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning video object segmentation with visual memory. In ICCV, 2017. 1, 2, 3, 6
[54] Roberto Vezzani, Davide Baltieri, and Ritu Cucchiara. People reidentification in surveillance and forensics: A survey. CSUR, 2013. 1
[61] Wenguan Wang, Jianbing Shen, and Ling Shao. Consistent video saliency using local gradient flow optimization and global refinement. TIP, 2015. 1, 2, 6, 7