A Generative Appearance Model for End-to-end Video Object Segmentation

Joakim Johnander[1,3]   Martin Danelljan[1,2]   Emil Brissman[1,4]   Fahad Shahbaz Khan[1,5]
Michael Felsberg[1]

[1] CVL, Linköping University, Sweden   [2] CVL, ETH Zürich, Switzerland   [3] Zenuity, Sweden   [4] Saab, Sweden   [5] IIAI, UAE

Abstract

One of the fundamental challenges in video object segmentation is to find an effective representation of the target and background appearance. The best performing approaches resort to extensive fine-tuning of a convolutional neural network for this purpose. Besides being prohibitively expensive, this strategy cannot be truly trained end-to-end since the online fine-tuning procedure is not integrated into the offline training of the network.

To address these issues, we propose a network architecture that learns a powerful representation of the target and background appearance in a single forward pass. The introduced appearance module learns a probabilistic generative model of target and background feature distributions. Given a new image, it predicts the posterior class probabilities, providing a highly discriminative cue, which is processed in later network modules. Both the learning and prediction stages of our appearance module are fully differentiable, enabling true end-to-end training of the entire segmentation pipeline. Comprehensive experiments demonstrate the effectiveness of the proposed approach on three video object segmentation benchmarks. We close the gap to approaches based on online fine-tuning on DAVIS17, while operating at 15 FPS on a single GPU. Furthermore, our method outperforms all previously published approaches on the large-scale YouTube-VOS dataset.

1. Introduction

Video object segmentation (VOS) is the task of tracking *and* segmenting one or multiple target objects in a video sequence. In this work, we consider the semi-supervised setting, where the ground-truth segmentation is only given in the first frame. The task is generic, i.e., the targets are arbitrary and no further assumptions regarding the object classes are made. The VOS problem is challenging from several aspects. The target may undergo significant appearance changes and may be subject to fast motion or occlusion. Moreover, the scene may contain distractor objects

Figure 1. Comparison between our proposed approach and the recently proposed RGMP [31]. In RGMP, the input features are concatenated with the initial mask and feature map. In contrast, we explicitly capture the target and background appearance, including distractor objects, by generative modelling. While RGMP severely struggles, the proposed approach successfully identifies and accurately segments all annotated targets. As in RGMP, we do not invoke computationally intensive fine-tuning in the first frame, but instead aim to learn the appearance model in a single forward pass. The figure is best viewed in colour.

that are visually or semantically similar to the target.

To tackle the aforementioned challenges, the standard strategy is to invoke extensive iterative optimization in the first frame [1, 2, 21, 30], given the initial image-mask pair. However, this strategy comes at an immense computational cost, rendering real-time operation infeasible. Furthermore, these methods do not train the segmentation pipeline end-to-end, since the online fine-tuning step is excluded from the offline learning stage. In response to the these issues, we explore the problem of finding a feedforward network architecture for VOS that completely avoids online optimization.

Recent works have posed video object segmentation as a feedforward mask-refinement process [23,31,34], where the previous mask prediction is adapted to fit the target in the current frame using a convolutional neural network. How-

ever, since no explicit modelling of the target appearance is performed, such approaches inherently fail if the target is occluded or out of view. This problem has been approached by incorporating simple appearance models based on e.g., concatenation of the feature map from the first frame [31], or utilization of a set of foreground and background feature vectors [4, 13]. However, these appearance models are either too simplistic, achieving unsatisfactory discriminative power, or cannot be fully trained end-to-end due to the reliance of non-differentiable components.

In this work, we propose a novel neural network architecture for video object segmentation that integrates a powerful appearance model of the scene. In contrast to previous methods, our network internally learns a generative probabilistic model of the foreground and background feature distributions. For this purpose, we employ a class-conditional mixture of Gaussians, which is inferred through a single forward pass. Our appearance model outputs the posterior class probabilities, thus providing a powerful cue containing discriminative information about the image content. This completely removes the need for online fine-tuning, as target-specific appearance information is captured in a single forward pass. We demonstrate our approach in fig. 1.

The proposed generative appearance model is seamlessly integrated as a module in our video object segmentation network. Our complete architecture is composed of a backbone feature extractor, the generative appearance module, a mask propagation branch, a fusion component, and a final upsampling and prediction module. For our generative appearance module, both the model inference and the prediction stages are fully differentiable. This ensures that the entire segmentation pipeline can be trained end-to-end, which is not the case for methods invoking online fine-tuning [1, 2, 12, 21, 23, 30] or K-Nearest-Neighbor prediction [4, 13]. Finally, our appearance module is lightweight, enabling efficient online inference.

We perform extensive experiments on 3 datasets, including the recent large-scale YouTubeVOS dataset [32]. We obtain a final score of 66.0% on YouTube-VOS, outperforming all previously published methods. Further, our approach achieves the best mean IoU of 67.2% on Davis17 among all causal video object segmentation methods. We perform a comprehensive analysis of our method in terms of an ablation study. Our analysis clearly underlines the effectiveness of the proposed generative appearance module and the importance of full end-to-end learning.

2. Related Work

In this work we address the problem of video object segmentation where an initial segmentation mask is provided, defining the target in the first frame. In recent years interest in this problem has surged and a wide variety of approaches have been proposed. Caelles et al. [2] proposed to use a convolutional neural network pre-trained for the semantic segmentation task, and fine-tune this in the first frame to segment out foreground and background. The approach was extended in a number of works: continuous training during the sequence [30]; adding instance-level semantic information [21]; incorporating motion information via optical flow [1, 6, 12]; performing temporal propagation via a Markov random field [1]; location-specific embeddings [8]; sophistic data augmentation [16]; or a combination of these [20]. While these approaches obtain satisfactory results in many scenarios, they have one critical drawback in common: they learn the target appearance in the initial frame via extensive training of deep neural networks with stochastic gradient descent. This leads to a significant time-delay before these methods can start tracking, and an average computation time that renders real-time processing infeasible.

Despite reduced accuracy, several approaches avoid invoking expensive fine-tuning procedures in the first frame. Some methods rely on optical flow coupled with refinement [15, 29]. Li et al. proposed DyeNet [18], which combines optical flow with an object proposal network, interleaving bidirectional mask-propagation and target re-identification. DyeNet provides outstanding performance, but it is not causal and relies on future video frames to make predictions. Jampani et al. [14] explicitly try to avoid optical flow and propose an approach based on bilateral filters. Cheng et al. [5] track different parts of the target with visual object tracking techniques, and refine the final solution with a convolutional neural network. Xu et al. [32] instead train a convolutional LSTM [11] to track and segment the target.

More closely related to our work, Perazzi et al. [23] pose video object segmentation as a mask refinement problem. Based on an input image, the mask predicted from the previous frame is refined with a neural network. The network is recurrent in time, with a particularly deep recurrent connection, an entire VGG16 [28]. In the work by Yang et al. [34], the mask was reduced to a rough spatial prior on the target location, and this together with a channel-wise attention mechanism provided improved performance. Wug et al. [31] extend [23] and concatenate the initial frame feature map and mask with the current feature map and previous mask, and train a standard convolutional neural network to match and segment in a fully recurrent fashion. Also more explicit matching mechanisms have been proposed, where the input features are matched with a set of features with known class membership [4, 13] using K-Nearest-Neighbour (KNN). While these methods model the target appearance, the non-parametric nature of KNN requires the entire training set to be stored. Additionally, the process of finding the K nearest neighbours is not differentiable. In contrast to existing work, our approach learns a compact appearance model of the scene in a single differentiable forward pass.

Figure 2. Full architecture of the proposed approach, illustrating both model initialization and frame processing. Model Initialization: A feature map is extracted from the initial frame, which is then fed together with the mask to the mask propagation module. This pair is furthermore used to initialize the appearance model. Frame processing: A feature map is extracted from the current frame and fed to both the appearance and mask-propagation modules whose outputs are combined, generating a coarse mask-encoding. Our upsampling module then refines the mask-encoding by also considering low-level information contained in the shallow features. The predictor then generates a final segmentation, based on this encoding. Moreover, the mask-encoding and appearance model parameters are fed back via a recurrent connection. During training, we use two cross-entropy losses applied to the coarse and fine segmentations, respectively.

3. Method

The aim of this work is to develop a network architecture for video object segmentation with the capability of learning accurate models of the target and background appearance through a single forward pass. That is, the network must learn in a one-shot manner to discriminate between target and background pixels, without invoking stochastic gradient descent. We tackle this problem by integrating a generative model of the foreground and background appearance. This model directly aids the segmentation process by providing discriminative posterior class probabilities. The learning and inference is computationally efficient and end-to-end differentiable, enabling a seamless integration of our generative component into a neural network.

3.1. Overview

Our approach is divided into five components that jointly address the video object segmentation task and are trained jointly end-to-end. The model is illustrated in fig. 2. Given an input image, features are first extracted with a backbone network. These are then passed to the appearance- and mask-propagation modules. The outputs of these two modules are combined in the fusion module, comprising two convolutional layers and outputting a coarse mask encoding. The encoding is handed to a predictor that generates a coarse segmentation mask. This prediction is used to update the appearance module and further used as input to the mask-propagation layer in the next frame to provide a rough spatial prior. The mask encoding output by the fusion component is also passed through an upsampling module, in which the coarse encoding is combined with successively more shallow features in order to produce a final refined segmentation.

3.2. Generative Appearance Module

The task of our appearance module is to learn a generative model of the video content in a deep feature space. Our generative model is *conditioned* on the class variable, indicating target or background. Given a new frame, the appearance module returns the posterior class probabilities at each image location. This output forms an extremely strong cue for foreground/background discrimination, as the proposed module explicitly models their respective appearance in a probabilistic manner.

Model learning: Formally, let the set of features extracted from the image be denoted as $\{\mathbf{x}_p\}_p$. The feature $\mathbf{x}_p$ at each spatial location p is a D-dimensional vector of real numbers. We model these observed feature vectors as i.i.d. samples drawn from the underlying distribution

$$p(\mathbf{x}_p) = \sum_{k=1}^{K} p(z_p = k) p(\mathbf{x}_p | z_p = k) \; . \qquad (1)$$

Each class-conditional density is a multi-variate Gaussian with mean $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$,

$$p(\mathbf{x}_p | z_p = k) = \mathcal{N}(\mathbf{x}_p | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \; . \qquad (2)$$

The discrete random variable z_p in (1) assigns the observation $\mathbf{x}_p$ to a specific component $z_p = k$. We use a uniform prior $p(z_p = k) = 1/K$ for this variable, where K is the number of components. Each component exclusively models the feature vectors of either the foreground or background. As further detailed below, we use four Gaussians, where the components $k \in \{0, 2\}$ model background and $k \in \{1, 3\}$ model foreground features.

In the first frame, our generative mixture model is inferred from the extracted features and the initial target mask. In subsequent frames, we update the model using the network predictions as soft class labels. In general, to update

the mixture model in a frame i we require a set of features $\mathbf{x}_p^i$ together with a set of soft component assignment variables $\alpha_{p,k}^i \in [0,1]$. These variables can be thought of as soft labels, describing the level of assignment of the vector $\mathbf{x}_p^i$ to component k. In the first frame $i = 0$, the feature vectors would be strictly assigned to either foreground or background $\alpha_{p,k}^0 \in \{0,1\}$, using the initial target mask.

Given the variables $\alpha_{p,k}^i$, we compute the model parameter updates as,

$$\tilde{\boldsymbol{\mu}}_k^i = \frac{\sum_p \alpha_{p,k}^i \mathbf{x}_p^i}{\sum_p \alpha_{p,k}^i} \ , \tag{3a}$$

$$\tilde{\boldsymbol{\Sigma}}_k^i = \frac{\sum_p \alpha_{p,k}^i \, \text{diag}\{(\mathbf{x}_p^i - \tilde{\boldsymbol{\mu}}_k^i)^2 + \mathbf{r}_k\}}{\sum_p \alpha_{p,k}^i} \ . \tag{3b}$$

For efficiency, we limit the covariance matrix to be diagonal, where $\text{diag}\{\mathbf{v}\}$ is a diagonal matrix with entries corresponding to the input vector $\mathbf{v}$. To avoid singularities, the covariance is regularized with a vector $\mathbf{r}_k$, which is a trainable parameter in our network. In the first frame, the mixture model parameters in (2) are directly achieved from (3), i.e. $\boldsymbol{\mu}_k^0 = \tilde{\boldsymbol{\mu}}_k^0$ and $\boldsymbol{\Sigma}_k^0 = \tilde{\boldsymbol{\Sigma}}_k^0$. In subsequent frames, these parameters are updated with new information (3) using a learning rate λ,

$$\boldsymbol{\mu}_k^i = (1-\lambda)\boldsymbol{\mu}_k^{i-1} + \lambda\tilde{\boldsymbol{\mu}}_k^i \ ,$$
$$\boldsymbol{\Sigma}_k^i = (1-\lambda)\boldsymbol{\Sigma}_k^{i-1} + \lambda\tilde{\boldsymbol{\Sigma}}_k^i \ . \tag{4}$$

Assignment variables: Next, we describe the computation of the assignment variables $\alpha_{p,k}^i$. Note that (3) resembles the M-step in the Expectation Maximization (EM) algorithm for a mixture of Gaussians. In EM, the variables z_p^i are treated as latent and (3) is derived by maximizing the expected complete-data log-likelihood. In that case the assignment variables are computed in the E-step as $\alpha_{p,k}^i = p(z_p^i = k|\mathbf{x}_p^i, \theta^{i-1})$, where $\theta^{i-1} = \{\boldsymbol{\mu}_k^{i-1}, \boldsymbol{\Sigma}_k^{i-1}\}_k$ are the previous estimates of the parameters. However, the setting is different in our case. The discrete assignment variables z_p^i are fully observed in the first frame. Moreover, in the subsequent frames, the network refines the posteriors $p(z_p^i = k|\mathbf{x}_p^i, \theta^{i-1})$, providing even better assignment estimates. We therefore exploit these factors in the estimation of the assignment variables α_{kp}^i.

Our model consists of one *base component* for background $k = 0$ and foreground $k = 1$, respectively. Given the ground truth binary target mask y_p in the first frame, where $y_p = 1$ for foreground and $y_p = 0$ otherwise, we set $\alpha_{p,0}^0 = 1 - y_p$ and $\alpha_{p,1}^0 = y_p$. That is, the feature vectors x_p^i are strictly assigned to the foreground and background base components according to the initial mask. In subsequent frames, where the ground-truth is not available, we use the

final prediction of our segmentation network according to

$$\alpha_{p,0}^i = 1 - \tilde{y}_p(I^i, \theta^{i-1}, \Phi)$$
$$\alpha_{p,1}^i = \tilde{y}_p(I^i, \theta^{i-1}, \Phi) \ . \tag{5}$$

Here, $\tilde{y}_p(I^i, \theta^{i-1}, \Phi) \in [0,1]$ is the probability of the target class, given the input image I^i, neural network parameters Φ, and current mixture model parameter estimates θ^{i-1}.

A drawback of using a single Gaussian component per class is that only uni-modal distributions can be accurately represented. However, the background appearance is typically multi-modal, especially in the presence of background objects that are similar to the target, often termed *distractors*. To obtain satisfactory discrimination between foreground and background, it is therefore critical to capture the feature distribution of such distractors. We therefore add Gaussian components in our model that are dedicated to the task of modeling hard examples. These components are explicitly learned to counter the errors of the two base components. Ideally, we would wish the base components alone to correctly predict the assignment variables, i.e. $p(z_p^i = k|\mathbf{x}_p^i, \boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i) = \alpha_{p,k}^i$, $k = 0, 1$. The additional components are trained on data where this does not hold by considering incorrectly classified background ($k = 2$) and foreground ($k = 3$) respectively. Their corresponding assignment variables are computed as,

$$\alpha_{p,2}^i = \max(\mathbf{0}, \alpha_{p,0}^i - p(z_p^i = 0|\mathbf{x}_p^i, \boldsymbol{\mu}_0^i, \boldsymbol{\Sigma}_0^i))$$
$$\alpha_{p,3}^i = \max(\mathbf{0}, \alpha_{p,1}^i - p(z_p^i = 1|\mathbf{x}_p^i, \boldsymbol{\mu}_1^i, \boldsymbol{\Sigma}_1^i)) \ . \tag{6}$$

The posteriors $p(z_p^i = k|\mathbf{x}_p^i, \boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i)$ are evaluated using only the base components. Given (6), we finally update the parameters of the components $k = 2, 3$ using (3) and (4).

Module output: Given the mixture model parameters computed in the previous frame, θ^{i-1}, our model can predict the component posteriors,

$$p(z_p^i = k|\mathbf{x}_p^i, \theta^{i-1}) = \frac{p(z_p^i = k)p(\mathbf{x}_p^i|z_p^i = k)}{\sum_k p(z_p^i = k)p(\mathbf{x}_p^i|z_p^i = k)} \ . \tag{7}$$

Note that each component k belongs to either foreground or background, and that the outputs (7) thus provide a discriminative mask encoding. In practice, we found it beneficial to feed the log-probabilities $\log(p(z_p^i = k)p(\mathbf{x}_p^i|z_p^i = k))$ into the conv layers in the fusion module. By canceling out constant factors, the outputs are calculated as,

$$s_{pk}^i = -\frac{\ln|\boldsymbol{\Sigma}_k^{i-1}| + (\mathbf{x}_p^i - \boldsymbol{\mu}_k^{i-1})^{\mathrm{T}}(\boldsymbol{\Sigma}_k^{i-1})^{-1}(\mathbf{x}_p^i - \boldsymbol{\mu}_k^{i-1})}{2} . \tag{8}$$

The component posteriors (7) can be reconstructed from s_{pk}^i by a simple soft-max operation. The output (8) should therefore be interpreted as component *scores*, encoding foreground and background assignment. The entire appearance modelling procedure is summarized in Algorithm 1.

3.3. Object Segmentation Architecture

As our backbone feature extractor, we use ResNet101 [10] with dilated convolutions [3] to reduce the stride of the deepest layer from 32 to 16. It is pretrained on ImageNet and all layers up to the last block, `layer4`, are frozen. The *mask-propagation module* is based on the concept proposed in [31]. The module constructs a mask encoding based on the mask predicted in the previous frame, a feature map predicted in the current frame, and a feature map extracted from the initial frame together with the given ground-truth mask. The entire module consists of three convolutional layers, where the middle layer is a dilation pyramid [3].

The outputs of the mask propagation and appearance modules are concatenated and fed into the *fusion module*, comprising two convolution layers. The result is processed by the *upsampling module* from which a predicted soft segmentation $\hat{y}_p$ is obtained. The output of the fusion module is also fed into a predictor that produces a coarse segmentation $\tilde{y}_p$, which is utilized by the mask propagation and appearance modules (using (5)) in the next timestep. By separating the feature extractor and upsampling path from the recurrent module we get a shorter path between variables of different time steps. We experienced the coarse mask to be a sufficient representation of the previous target segmentation. As a special case, during sequences with multiple objects, we run our approach once per object and combine the resulting soft segmentations with softmax-aggregation [31]. The aggregated soft segmentations then replaces the coarse segmentations $\tilde{y}_p$ in the recurrent connection.

The output of the fusion module provides coarse mask-encoding that is used to locate and segment the target. There have been considerable efforts in semantic segmentation and instance segmentation litterature to refine final segmentations. We adopt an upsampling path similar to [25], where the coarse representation is successively combined with successively shallower features.

3.4. Network Training

We train the proposed neural network end-to-end in a recurrent fashion. Based on a video and a single ground-truth segmentation, the network predicts segmentation masks for each frame in the video. We train on three datasets:

DAVIS2017 [26]: The DAVIS2017 training set comprises 60 videos containing one or several annotated objects to track. Each video is between 25 and 100 frames long, each of which is labeled with a ground-truth segmentation.

YouTube-VOS [32]: The YouTube-VOS training set consists of 3471 videos with one or several target objects. Each video is 20 to 180 frames long, where every fifth frame is labelled. We use only the labelled frames during training.

SynthVOS: In order to cover a wide varity of classes we follow [23, 31] and utilize objects from the salient object segmentation dataset MSRA10k [7]. It contains 10^4 images

---

Algorithm 1: The appearance module inference and update. **Inference:** Based on the appearance model parameters, $\boldsymbol{\mu}_k^i$, $\boldsymbol{\Sigma}_k^i$, and the input feature map $\mathbf{x}_p^i$, a soft segmentation is constructed for the background, foreground, and the two residual components. **Update:** The appearance model parameters are updated based on the coarse segmentation $\tilde{y}_p^i$.

---

1 Inference($\mathbf{x}_p^i$, $\boldsymbol{\mu}_k^i$, $\boldsymbol{\Sigma}_k^i$):
2   for $k = 0, 1, 2, 3$: compute s_{pk}^i from (8)
3   **return** s_{pk}^i
4 Update($\mathbf{x}_p^i$, $\tilde{y}_p^i$, $\boldsymbol{\mu}_k^i$, $\boldsymbol{\Sigma}_k^i$):
5   for $k = 0, 1$: compute $\alpha_{p,k}^i$ from (5)
6   for $k = 0, 1$: compute $\tilde{\boldsymbol{\mu}}_k^i$, $\tilde{\boldsymbol{\Sigma}}_k^i$ based on (3)
7   for $k = 0, 1$: compute s_{pk}^i based on (8)
8   for $k = 0, 1$: compute
        $p(z_p^i = k | \mathbf{x}_p^i, \boldsymbol{\mu}_0^i, \boldsymbol{\Sigma}_0^i) = \text{Softmax}(s_{p0}^i, s_{p1}^i)$
9   for $k = 2, 3$: compute $\alpha_{p,k}^i$ from (6)
10   for $k = 2, 3$: compute $\tilde{\boldsymbol{\mu}}_k^i$, $\tilde{\boldsymbol{\Sigma}}_k^i$ based on (3)
11   for $k = 0, 1, 2, 3$: update $\boldsymbol{\mu}_k^i$ and $\boldsymbol{\Sigma}_k^i$ from (4)
12   **return** $\boldsymbol{\mu}_k^i$ and $\boldsymbol{\Sigma}_k^i$

---

where a single object is segmented. We paste 1 to 5 such objects onto an image from VOC2012 [9]. A synthetic video is obtained by moving the objects across the image.

One training sample consists of a video snippet of n frames and a given ground-truth for the first frame. Images are normalized with ImageNet [27] mean and standard deviation. We let our model predict segmentation masks in each frame and apply a cross-entropy loss. We also place an auxillary loss on the coarse segmentations $\tilde{y}_p$. The losses are summed and minimized with Adam in two stages:

Initial training: First we train for 80 epochs using all three datasets on half resolution images (240×432). The batch-size is set to 4 video snippets, using 8 frames in each snippet. We use a learning rate of 10^{-4}, exponential learning rate decay of 0.95 per epoch, and a weight decay of 10^{-5}.

Finetuning: We then finetune for 100 epochs on the DAVIS2017 and YouTube-VOS training sets, using full image resolution. During this step we sample sequences from both datasets with equal probability. The batchsize is lowered to 2 snippets, to accomodate longer sequences of 14 frames. We use a learning rate of 10^{-5}, exponential learning rate decay of 0.985 per epoch, and a weight decay of 10^{-6}. The training is stopped early by observing the performance on a held-out set of 300 sequences from the YouTube-VOS training set.

4. Experiments

We first conduct an ablation study of the proposed approach on the Youtube-VOS benchmark [32]. Then,

we compare with the state-of-the-art on three video object segmentation datasets [24, 26, 32]. Our approach, called **A-GAME**, is implemented in PyTorch [22] and trained on a single Nvidia V100 GPU. Our code and trained networks are available at https://github.com/joakimjohnander/agame-vos.

4.1. Ablation Study

We perform an extensive ablative analysis of our approach on the large-scale YouTube-VOS dataset. We use the official validation set, comprising 474 videos labelled with one or multiple objects. Ground-truth masks are withheld, and results are obtained through an online evaluation server. Performance is measured in terms of the mean Jaccard index $\mathcal{J}$ [26], i.e. intersection-over-union (IoU), and the mean contour accuracy $\mathcal{F}$. The two measures are separately calculated for seen and unseen classes, resulting in four performance measures. The overall performance ($\mathcal{G}$) is the average of all four measures.

In our ablative experiments, we analyze six key modifications of our approach, as explained below. Results are shown in table 1. For each version, we retrain the entire network from scratch using the exact same procedure.

Appearance module: We first analyze the impact of the proposed appearance module (see section 3.2) by removing it from the network (*No appearance module* in table 1). This leads to a major reduction in overall performance, from 66.0% to 50.0%. The results clearly demonstrate that the introduced appearance module is an essential component in our video object segmentation approach. Further insights are obtained by studying the performance on *seen* and *unseen* classes in table 1. Note that removing the appearance module causes a 9.1% decrease for classes that are seen during training, and a remarkable 20.6% decrease for unseen classes. Thus, our generative appearance model component is crucial for the generalization to arbitrary objects that are unseen during training. This is explained by the target specific and class-agnostic nature of our appearance module.

Mask-propagation module: Secondly, we investigate the importance of the mask-propagation module (see section 3.3). Refraining from propagating the mask predicted

Version	$\mathcal{G}$	$\mathcal{J}$ seen (%)	$\mathcal{J}$ unseen (%)
A-GAME	66.0	66.9	61.2
No appearance module	50.0	57.8	40.6
No mask-prop module	64.0	65.5	59.5
Unimodal appearance	64.4	65.8	58.8
No update	64.9	66.0	59.8
Appearance SoftMax	55.8	59.3	50.7
No end-to-end	58.8	62.5	53.1

Table 1. Ablation study on YouTube-VOS. We report the overall performance $\mathcal{G}$ along with segmentation accuracy $\mathcal{J}$ on classes seen and unseen during training. See text for further details.
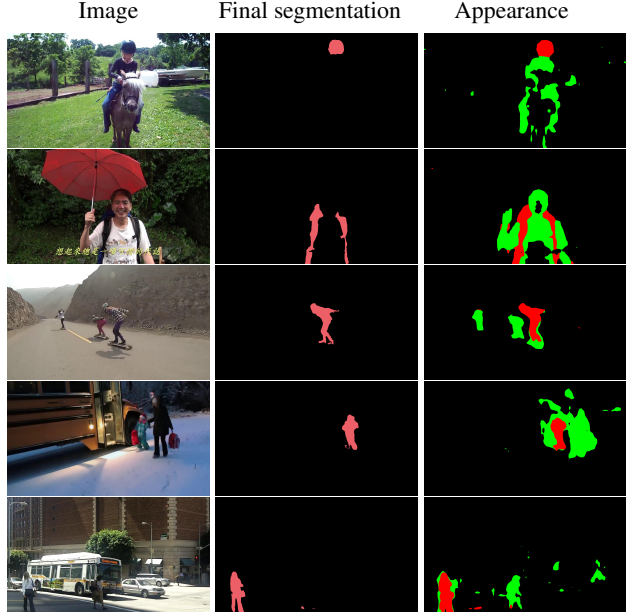

Figure 3. Visualization of the appearance module on five videos from YouTube-VOS. The final segmentation of our approach is shown (middle) together with output of the appearance module (right). The appearance module accurately locates the target (red) with the foreground representation while accentuating potential distractors (green) with the secondary mixture component.

in the previous frame (*No mask-prop module* in table 1) results in a 2.0% reduction in performance. While this reduction is significant, the importance of the mask-propagation module is small compared to that of the appearance module.

Gaussian mixture components: As described in section 3.2, we employ two Gaussian mixture components to model the foreground and background, respectively. In addition to the base mixture component, a secondary Gaussian mixture component is added to capture hard examples that are not accurately modelled by a unimodal distribution. We investigate the impact of this additional mixture components by removing them from our model. The resulting version (*Unimodal appearance* in table 1) thus only employs a single base mixture component for each class. The resulting performance drop of 1.6% indicates the importance of modeling hard examples in the presence of distractor objects.

The impact of the multi-modal generative model is also analyzed qualitatively in fig. 3. The mixture component dedicated to hard negative image regions is able to model other objects in the vicinity of the target (row 1 and 2) and accurately captures other objects of the same class (row 3-5). Note that both the appearance module's output and the final segmentations are soft, and only for the purpose of visualization we show the arguments of the maxima.

Model update: We investigate the impact of updating the generative model in each frame using (4). The version *No update* (table 1) only uses the initial frame to compute the mixture model parameters (3), and no update (4) is per-

Method	O-Ft	$\mathcal{G}$ overall (%)	$\mathcal{J}$ seen (%)	$\mathcal{J}$ unseen (%)
S2S [33]	✓	64.4	71.0	55.5
OSVOS [2]	✓	58.8	59.8	54.2
OnAVOS [30]	✓	55.2	60.1	46.6
MSK [23]	✓	53.1	59.9	45.0
OSMN [34]	✗	51.2	60.0	40.6
S2S [33]	✗	57.6	66.7	48.2
RGMP [31]	✗	53.8	59.5	45.2
RGMP† [31]	✗	50.5	54.1	41.7
A-GAME	✗	66.0	66.9	61.2
A-GAME†	✗	66.1	67.8	60.8

Table 2. State-of-the-art comparison on the YouTubeVOS benchmark. Our approach obtains the best overall performance ($\mathcal{G}$) despite not performing any online fine-tuning (O-Ft). Further, our approach provides a large gain in performance for categories unseen during training ($\mathcal{J}$ unseen), compared to existing methods. Entries marked by † were trained with only YouTube-VOS data.

formed during training and inference. Updating the generative model to capture changes in the target and background appearance leads to a 1.1% improvement in performance.

Appearance module output: As previously described, our appearance module outputs the log-probability scores (8). To validate this choice, we also compare with outputting the posterior probabilities (*Appearance SoftMax* in table 1), obtained by adding a `SoftMax` layer after computing the scores (8), between the appearance and fusion modules. This leads to a significant degradation in performance (-10.2%). These results are in line with conventional techniques in segmentation [19] and classification [17], where activations in the network are not converted to probabilities until the final output layer.

End-to-end learning: Finally, we analyze the impact of end-to-end differentiation and training in our approach. Specifically, we investigate the importance of end-to-end differentiability in the learning stage of the appearance module. The comparison is performed by not backpropagating through the model inference computation (3) during the training of the network. Note that, the rest of the framework remains unchanged. The resulting method (*No end-to-end* in table 1) obtains poor results, with a total degradation of 7.2% in overall performance. This highlights the importance of permitting *true* end-to-end learning.

4.2. State-of-the-Art Comparison

We compare our approach with the state-of-the-art on three video object segmentation benchmarks: YouTube-VOS [32], DAVIS2017 [26], and DAVIS2016 [24].

YouTube-VOS: This recently introduced large-scale dataset contains 474 sequences with 91 classes, 26 of which are not included in the YouTube-VOS training set. We use the official validation set, as in section 4.1. We compare our approach with all, to our best knowledge, published results [32]. Additionally, we evaluate the RGMP method, using the code provided by the authors. The results are shown in table 2. For each approach, we indicate if the method employs online fine-tuning (O-Ft) and if it is causal,

Method	O-Ft	Causal	$\mathcal{J}\&\mathcal{F}$ mean (%)	$\mathcal{F}$ (%)	$\mathcal{J}$ (%)
CINM [1]	✓	✓	70.6	74.0	67.2
OSVOS-S [21]	✓	✓	68.0	71.3	64.7
OnAVOS [30]	✓	✓	65.4	69.1	61.6
OSVOS [2]	✓	✓	60.3	63.9	56.6
DyeNet [18]	✗	✗	69.1	71.0	67.3
RGMP [31]	✗	✓	66.7	68.6	64.8
VM [13]	✗	✓	-	-	56.5
FAVOS [5]	✗	✓	58.2	61.8	54.6
OSMN [34]	✗	✓	54.8	57.1	52.5
A-GAME	✗	✓	70.0	72.7	67.2

Table 3. State-of-the-art comparison on the DAVIS2017 validation set. For each method we report whether it employs online fine-tuning (O-Ft), is causal, and the final performance $\mathcal{J}$ (%). Our approach obtains superior results compared to state-of-the-art methods without online fine-tuning. Further, our approach closes the performance gap to existing methods employing online fine-tuning.

Method	O-Ft	Causal	Speed	$\mathcal{J}\&\mathcal{F}$ mean (%)	$\mathcal{F}$ (%)	$\mathcal{J}$ (%)
OnAVOS [30]	✓	✓	13s	85.5	84.9	86.1
OSVOS-S [21]	✓	✓	4.5s	86.6	87.5	85.6
MGCRN [12]	✓	✓	0.73s	85.1	85.7	84.4
CINM [1]	✓	✓	>30s	84.2	85.0	83.4
LSE [8]	✓	✓		81.5	80.1	82.9
OSVOS [2]	✓	✓	9s	80.2	80.6	79.8
MSK [23]	✓	✓	12s	77.6	75.4	79.7
SFL [6]	✓	✓	7.9s	75.4	76.0	74.8
DyeNet [18]	✗	✗	0.42s	-	-	84.7
FAVOS [5]	✗	✓	1.80s	81.0	79.5	82.4
RGMP [31]	✗	✓	0.13s	81.8	82.0	81.5
VM [13]	✗	✓	0.32s	-	-	81.0
MGCRN [12]	✗	✓	0.36s	76.5	76.6	76.4
PML [4]	✗	✓	0.28s	81.2	79.3	75.5
OSMN [34]	✗	✓	0.14s	73.5	72.9	74.0
CTN [15]	✗	✓	1.30s	71.4	69.3	73.5
VPN [14]	✗	✓	0.63s	67.9	65.5	70.2
MSK [23]	✗	✓	0.15s	-	-	69.9
A-GAME	✗	✓	0.07s	82.1	82.2	82.0

Table 4. State-of-the-art comparison on DAVIS2016 validation set, which is a subset of DAVIS2017. For each method we report whether it employs online fine-tuning (O-Ft), is causal, the computation time (if available), and the final performance $\mathcal{J}$ (%). Our approach obtains competitive results compared to causal methods without online fine-tuning.

i.e. if the segmentation output depends on future frames in the video. Here we let ✓ and ✗ indicate yes and no, respectively. Among previous approaches performing extensive online fine-tuning in the first frame, OSVOS and OnAVOS achieve final scores of 58.8% and 55.2%. For the S2S method, we compare with two versions: one with and one without online fine-tuning, obtaining 64.4% and 57.6%, respectively. Our approach obtains a final score of 66.0%, significantly outperforming state-of-the-art without invoking any online fine-tuning. Furthermore, our method performs notably well on the *unseen* category, which only considers objects that are not seen during training. Again, this demonstrates the effectiveness of our class-agnostic appearance module, which generalizes to arbitrary target objects.

DAVIS2017: The dataset comprises 30 videos with one or multiple target objects. The results are shown in table 3. Among existing methods, DyeNet is the only ap-

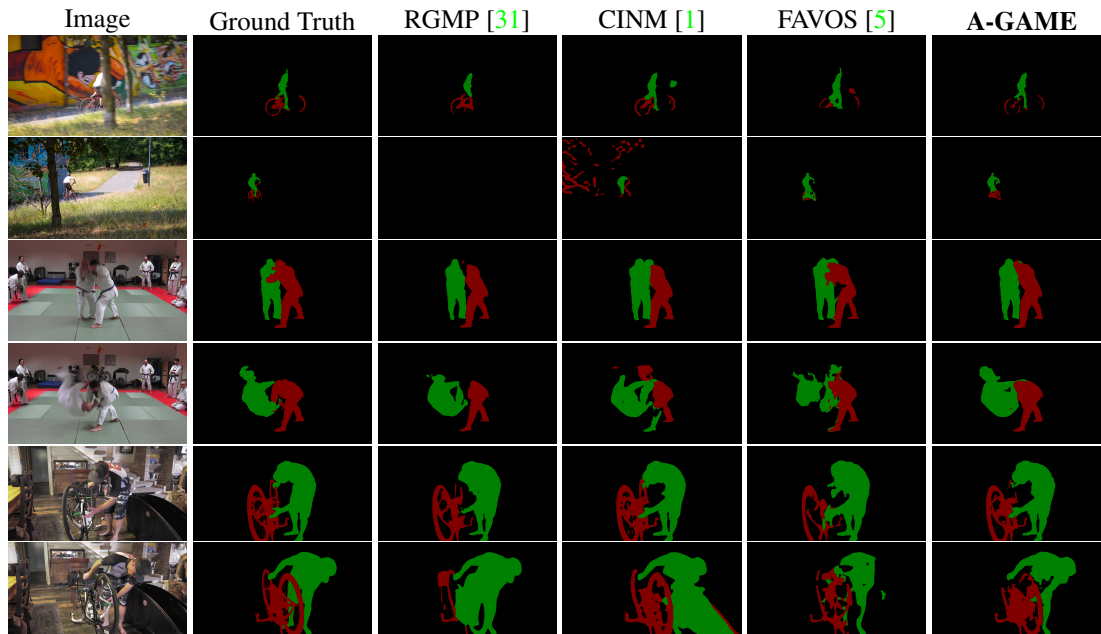| Image	Ground Truth	RGMP [31]	CINM [1]	FAVOS [5]	**A-GAME**

Figure 4. Qualitative comparison between our approach and 3 state-of-the-art approaches. Our approach is able to accurately segment all targets, demonstrating robustness to occlusions and successfully discriminating between different objects. This is largely thanks to the powerful appearance model in our architecture.

proach that is non-causal, since it processes the entire video in a bidirectional manner. It is therefore not applicable to real-time or online systems. The RGMP method, achieving a score of 64.8%, relies on mask propagation and an appearance model constructed by simply concatenating image features from the first frame. VideoMatch (VM) stores foreground and background feature vectors that are then matched with feature vectors in the test image. This method obtains a final result of 56.5%. The proposed method, employing an end-to-end differentiable generative probabilistic appearance model, achieves a score of 67.2%. Our approach outperforms all causal methods not invoking online fine-tuning, and is even on par with the best non-causal and online fine-tuning-based techniques.

DAVIS2016: For completeness, we also evaluate our approach on DAVIS2016. It is a subset of DAVIS2017, containing 20 videos labeled with a single object. The small size and number of objects in DAVIS2016 limits the diversity. It has therefore become highly saturated over the years. In table 4 we show the final result of each method, along with computational time reported by the respective authors. Our approach obtains a competitive performance of 82.0% compared to state-of-the-art. Unlike our method, the top performing approaches on DAVIS2016, such as OSVOS, OnAVOS, and FAVOS do not generalize well to the larger and more diverse YouTube-VOS and DAVIS2017 datasets.

4.3. Qualitative Evaluation

We qualitatively compare our approach with three state-of-the-art approaches (RGMP [31], CINM [1], FAVOS [5])

on three videos from DAVIS2017. The results are shown in fig. 4. RGMP tends to lose parts of objects, and struggles with discrimination between different objects. While CINM can produce detailed segmentation masks (row 5), it suffers from several failure modes (row 2, 4, 6). FAVOS struggles with discriminating targets (row 2, 6) and fails to capture details (row 6) or precise boundaries (row 4). The proposed approach succeeds to accurately segment both targets in all scenarios while being one or several orders of magnitude faster compared to FAVOS and CINM, respectively.

5. Conclusion

We propose to address the VOS problem by learning the appearance of the target in an efficient and differentiable manner, avoiding the drawbacks of existing matching or online-finetuning based approaches. The target appearance is modelled as a mixture of Gaussians in an embedding space, and we show that both learning and inference of this model can be expressed in closed form. This permits the implementation of the appearance model as a component in a neural network that is trained on end-to-end. We thoroughly analyze the proposed approach and demonstrate its effectiveness on three benchmarks, resulting in state-of-the-art performance.

References

[1] L. Bao, B. Wu, and W. Liu. Cnn in mrf: Video object segmentation via inference in a cnn-based higher-order spatio-temporal mrf. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5977–5986, 2018. 1, 2, 7, 8

[2] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. One-shot video object segmentation. In *CVPR 2017*. IEEE, 2017. 1, 2, 7

[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018. 5

[4] Y. Chen, J. Pont-Tuset, A. Montes, and L. Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1189–1198, 2018. 2, 7

[5] J. Cheng, Y.-H. Tsai, W.-C. Hung, S. Wang, and M.-H. Yang. Fast and accurate online video object segmentation via tracking parts. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2, 7, 8

[6] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang. Segflow: Joint learning for video object segmentation and optical flow. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 686–695. IEEE, 2017. 2, 7

[7] M. Cheng. Msra10k database, 2015. 5

[8] H. Ci, C. Wang, and Y. Wang. Video object segmentation by learning location-sensitive embeddings. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 501–516, 2018. 2, 7

[9] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015. 5

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[11] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 2

[12] P. Hu, G. Wang, X. Kong, J. Kuen, and Y.-P. Tan. Motion-guided cascaded refinement network for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1400–1409, 2018. 2, 7

[13] Y.-T. Hu, J.-B. Huang, and A. G. Schwing. Videomatch: Matching based video object segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 54–70, 2018. 2, 7

[14] V. Jampani, R. Gadde, and P. V. Gehler. Video propagation networks. In *Proc. CVPR*, volume 6, page 7, 2017. 2, 7

[15] W.-D. Jang and C.-S. Kim. Online video object segmentation via convolutional trident network. In *CVPR*, volume 1, page 7, 2017. 2, 7

[16] A. Khoreva, R. Benenson, E. Ilg, T. Brox, and B. Schiele. Lucid data dreaming for object tracking. In *The 2017 DAVIS Challenge on Video Object Segmentation - CVPR Workshops*, 2017. 2

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 7

[18] X. Li and C. Change Loy. Video object segmentation with joint re-identification and attention-aware mask propagation. In *The European Conference on Computer Vision (ECCV)*, September 2018. 2, 7

[19] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 7

[20] J. Luiten, P. Voigtlaender, and B. Leibe. Premvos: Proposal-generation, refinement and merging for the davis challenge on video object segmentation 2018, 2018. 2

[21] K. Maninis, S. Caelles, Y. Chen, J. Pont-Tuset, L. Leal-Taixe, D. Cremers, and L. Van Gool. Video object segmentation without temporal information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 1, 2, 7

[22] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017. 6

[23] F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung. Learning video object segmentation from static images. In *Computer Vision and Pattern Recognition*, volume 2, 2017. 1, 2, 5, 7

[24] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 724–732, 2016. 6, 7

[25] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár. Learning to refine object segments. In *European Conference on Computer Vision*, pages 75–91. Springer, 2016. 5

[26] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017. 5, 6, 7

[27] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, pages 1–42, April 2015. 5

[28] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 2

[29] Y.-H. Tsai, M.-H. Yang, and M. J. Black. Video segmentation via object flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3899–3908, 2016. 2

[30] P. Voigtlaender and B. Leibe. Online adaptation of convolutional neural networks for video object segmentation. In *British Machine Vision Conference 2017, BMVC 2017, London, UK, September 4-7, 2017*, 2017. 1, 2, 7

[31] S. Wug Oh, J.-Y. Lee, K. Sunkavalli, and S. Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *Proceedings of the IEEE Conference on Computer*

Vision and Pattern Recognition, pages 7376–7385, 2018. 1, 2, 5, 7, 8

[32] N. Xu, L. Yang, Y. Fan, J. Yang, D. Yue, Y. Liang, B. L. Price, S. Cohen, and T. S. Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part V*, pages 603–619, 2018. 2, 5, 6, 7

[33] N. Xu, L. Yang, Y. Fan, D. Yue, Y. Liang, J. Yang, and T. S. Huang. Youtube-vos: A large-scale video object segmentation benchmark. *CoRR*, abs/1809.03327, 2018. 7

[34] L. Yang, Y. Wang, X. Xiong, J. Yang, and A. K. Katsaggelos. Efficient video object segmentation via network modulation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1, 2, 7