MAIN: Multi-Attention Instance Network for Video Segmentation

Juan León Alcázar[*1], María A. Bravo[*1], Ali K. Thabet[2], Guillaume Jeanneret[1], Thomas Brox[3],

Pablo Arbeláez[1] and Bernard Ghanem[2]

[1]Universidad de los Andes, [2]King Abdullah University of Science and Technology, [3]University of Freiburg

[1]{jc.leon, ma.bravo641, g.jeanneret10, pa.arbelaez}@uniandes.edu.co;

[2]{ali.thabet, bernard.ghanem}@kaust.edu.sa;

[3]brox@cs.uni-freiburg.de

Abstract

Instance-level video segmentation requires a solid integration of spatial and temporal information. However, current methods rely mostly on domain-specific information (online learning) to produce accurate instance-level segmentations. We propose a novel approach that relies exclusively on the integration of generic spatio-temporal attention cues. Our strategy, named Multi-Attention Instance Network (MAIN)*, overcomes challenging segmentation scenarios over arbitrary videos without modelling sequence- or instance-specific knowledge. We design MAIN to segment multiple instances in a single forward pass, and optimize it with a novel loss function that favors class agnostic predictions and assigns instance-specific penalties. We achieve state-of-the-art performance on the challenging Youtube-VOS dataset and benchmark, improving the unseen Jaccard and F-Metric by 6.8% and 12.7% respectively, while operating at real-time (30.3 FPS).*

1. Introduction

Current state-of-the-art video segmentation methods [5, 48, 24, 30] have achieved impressive results for the binary task of separating foreground objects from the background. However, the finer-grained task of multi-instance video segmentation, which aims at independently identifying and segmenting multiple objects, remains an open research problem. There are several task-specific challenges for multi-instance segmentation. First, an accurate label assignment must create a set of spatial and temporal consistent masks across all the instances in the sequence. Second, there is a loose definition (lack of semantics) for the object of interest, since segmentation targets are arbitrarily chosen among all the available objects in the starting frame. Third, the appearance, scale, and visibility of segmentation targets vary throughout the video. Finally, there are complex scene dynamics specific to each sequence, which are

Figure 1. **MAIN video segmentation results.** These videos from the Youtube-VOS dataset present several challenges: large visual similarity, overlap and direct interaction between instances, and high variability in viewpoint and visual appearance. To visualize the animated figure, use Adobe Acrobat Reader.

hard to model without domain-specific knowledge. Figure 1 shows video clips with some results of our method on such challenging scenarios.

One-shot video segmentation defines the task of segmenting arbitrary objects given a ground-truth annotation in the first video frame. Methods like [5, 48, 24] approached this task by training a binary model over fully annotated videos (a phase commonly known as *Offline Training*) and finetuning it on every unseen video, resulting in multiple sequence-specific or instance-specific models (a phase also known as *Online Training*). These methods rely strongly on online training to estimate and fuse instance-level predictions. Recent methods have proposed other strategies to improve the online phase by using instance re-identification [43, 31, 7], instance proposals [54, 19, 44], or local motion information [47], among many others. Regardless of the selected strategy, these methods remain computationally expensive for training and inference, and might not be suitable for modern large-scale datasets.

In this paper, we propose a single encoder-decoder architecture that operates by design at the instance-level in
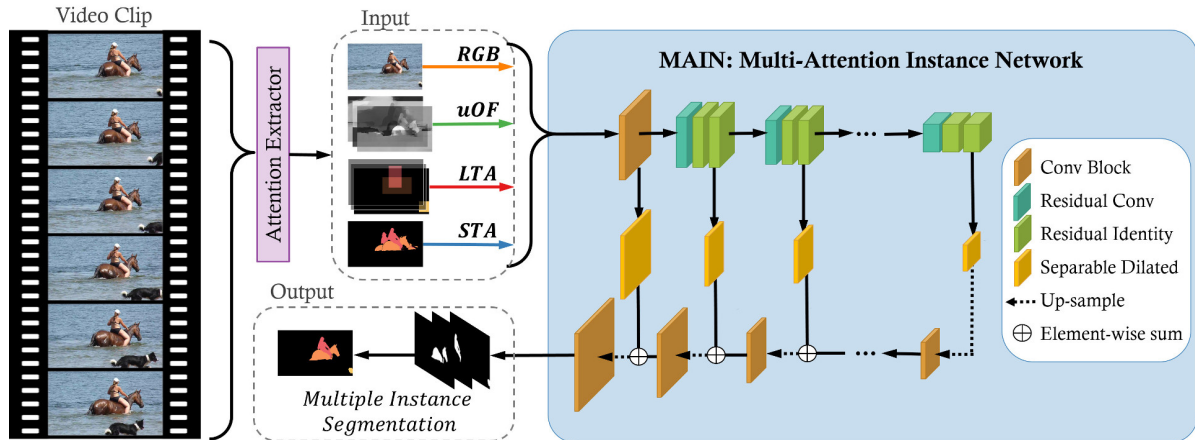
---

*Equal contribution

Figure 2. **Multi-Attention Instance Network (MAIN).** Our architecture integrates static cues, motion cues, long and short temporal cues in a single network to produce a multi-instance segmentation. We take as input the current RGB image, the Optical Flow (a robust version of the optical flow), and the Long and Short spatio-temporal attention (LTA and STA) cues and use them as input to our network. The decoder in MAIN uses multiple side-outputs that are combined following a Feature Pyramid Architecture of separable dilated convolutions.

an offline configuration, enabling efficient and generic processing of unconstrained video sequences. Our method focuses on recurrent estimation and consistent propagation of *generic spatio-temporal information*. It integrates static grouping features (*e.g.* color, shape and texture), with short-term motion cues (*e.g.* optical flow), along with short and long spatio-temporal attention mechanisms, which attend to a specific set of objects in a frame and maintain large contextual information. We optimize our model end-to-end using a novel loss function that assigns a pixel-wise penalty according to the relative error contribution of the segmentation target. Figure 2 shows an overview of our method, which we call *Multi-Attention Instance Network* (MAIN).

Intuitively, MAIN incorporates insights from multi-target tracking, image warping, and static image segmentation in an efficient framework that runs at 30.3 fps. To the best of our knowledge, MAIN is the first method that can generate an arbitrary number of instance-level predictions in a single forward pass and without domain-specific information, thus, making it particularly useful for unconstrained large-scale scenarios. MAIN accomplishes state-of-the-art performance and improves with a high increase for unseen instances in the Jaccard and F-Metric to 55.0% and 63.0% respectively. We verify our contributions through a series rigorous ablation studies in which we demonstrate the contribution of each additional feature and the robustness and effectiveness of our method.

Contributions. Our work has four main contributions. **(1)** MAIN directly addresses the multi-instance scenario; it generates multi-instance segmentations in a single forward pass, without the need of domain-specific knowledge or instance-specific fine-tuning. **(2)** We introduce a novel loss function devised for the multi-instance segmentation scenario. This function enables us to work on large and highly imbalanced datasets that contain multiple instances of diverse sizes without any hyper-parameter tuning. **(3)** MAIN explicitly fuses static, motion, short-term and long-term temporal grouping cues into a single end-to-end trainable architecture. **(4)** We propose a dilated separable decoder architecture, which allows us to aggregate multi-scale information using less computationally expensive operations.

To ensure reproducible results and to promote future research, all the resources of this project –source code, model weights, and official benchmark results– will be made publicly available.

2. Related Work

Interest in video segmentation has grown in recent years within the computer vision community, in part due to the availability of new datasets [40, 41, 53], and the development of deep convolutional neural architectures [16, 28, 27, 17, 35]. But mostly due to the recent re-formulation of the problem, evolving from a direct extension of classic image segmentation [3, 45, 13] into a one-shot learning task [40, 41, 53], which further highlights the need for temporally consistent estimations.

One-Shot learning for video object segmentation. Recent datasets formulate the video object segmentation task as a one-shot problem [40, 41, 53] and provide the ground-truth instance segmentations for the first frame. State-of-the-art methods [5, 39, 54, 48, 51] usually train a binary off-line model, capable of producing an initial background and foreground estimation. Then, during online training, they fine-tune these models for the specific instances in validation using the available ground-truth. Most offline methods [5, 46, 48, 24] do not work in a multi-instance scenario and require either multiple online training sessions or instance-specific knowledge [54] to do so. In contrast, MAIN works

directly on a multi-instance offline configuration, generating multi-instance segmentations in a single forward pass without any prior knowledge or online training.

Loss functions for video segmentation. Video object segmentation datasets have an inherent large class imbalance favoring trivial background assignments and making small objects into hard false negatives. To tackle this issue, strategies like the Focal Loss [33] use a sample-specific weight enabling the large imbalance to be controlled on detection tasks. Likewise, other methods [52, 11, 36] create an adaptive loss function for edge detection weighted by the ratio of positive edge pixels to background pixels. In the segmentation field, Milletari *et al.* [37] proposed the Dice Loss function, which directly approximates the Jaccard Index. In this paper, we propose a novel loss function, the Weighted Instance Dice (WID) loss, that exploits the benefits of the Dice loss to perform effectively on highly imbalanced datasets. WID independently weighs every pixel prediction according to the size of the corresponding instance.

Long-term temporal cues. Building upon the core ideas of generic and specific networks, some recent works [46, 47, 51] tackle the lack of consistent temporal information, a phenomenon that arises when video frames are processed independently. These approaches focus mostly on extracting useful temporal information from moving objects. In fact, motion is a strong bottom-up cue for video segmentation, especially when objects of interest have independent motion patterns [23]. Following this line of thought, some approaches [4, 50] use long-term point trajectories based on dense optical flow fields. They define pair-wise distances between these trajectories and cluster them to have temporally consistent segmentations of moving objects. We build upon these core ideas and design an encoder-decoder architecture that incorporates long-term spatial attention cues estimated from a tracking algorithm.

Short-term temporal cues. Recent video segmentation methods directly rely on motion information, thus including it as prior knowledge [24, 19, 39, 18, 51, 29]. These strategies either rely on dense optical flow estimation or another online approximation of pixel-level short-term motion patterns, either pre-computed or estimated jointly with the segmentation [8]. Compared to these methods, MAIN benefits from motion information by explicitly fusing a robust version of the optical flow with the standard RGB input.

Another important source of information when performing a recurrent task like video segmentation is the set of previous predictions. Methods like VPN [22, 7] and Mask-Track [39] use previously segmented masks to better estimate the current segmentation. We also use the optical flow to warp the previous segmentation and to set our short spatio-temporal prior information. MAIN fuses these short-term cues with long spatio-termporal cues to produce a set of temporally consistent instance segmentations.

3. Multi-Attention Instance Network (MAIN)

Video object segmentation datasets such as Youtube-VOS [53] and DAVIS-17 [41] contain multiple segmentation targets in a single video. Our approach directly addresses the multi-instance scenario with an offline trained network that produces multiple segmentations in a single forward pass. We concatenate attention sources for every instance in the video and optimize MAIN in order to output a segmentation only if the attention cues indicate the presence of an instance. Given a dataset with at most N instances per video, we use $2M$ attention priors for M possible instances (one short-term (STA) and one long-term (LTA) for each instance), and set the output of our decoder to return a tensor of dimensions $N \times H \times W$. Then, if a video has $M \leq N$ instances, MAIN predicts instances only for the first M channels.

While simple, this extension overcomes two important challenges of the one-shot multi-instance segmentation scenario: (i) the lack of semantics for target objects, since attention cues are class agnostic and (ii) the need for domain-specific knowledge to achieve temporally consistent instance-level segmentations, given by the multiple channel output that generates multi-instance segmentation in a single forward pass. By using this strategy, MAIN reduces the computational complexity of the forward pass, along with the total training time.

Instance Shuffle. We observe that the instance distribution in Youtube-VOS (Avg. 1.71, Std. 0.87, Mode 1, Max 6 per video) makes it difficult for the network to generate accurate multi-instance segmentations for videos with more than three instances, since they are not frequent. We address this problem by randomly shuffling the attention channels, and performing the same permutation in the output maps and supervision data. After this modification, instances appear with similar frequency across output channels in a single batch regardless of the bias in the dataset.

3.1. Weighted Instance Dice (WID) Coefficient Loss

Most state-of-the-art methods for video object segmentation use either a binary or a multi-class cross-entropy loss function that can, optionally, be weighted in the presence of a large class imbalance [36, 52]. Since MAIN predicts instance-level and class-agnostic segmentations, we depart from the standard practice and introduce a novel loss function that better approximates the multi-instance segmentation scenario. This loss penalizes overlapping predictions and errors over small instances, which have large influence on the evaluation metric. We propose the Weighted Instance Dice Coefficient (*WID*) loss function for a multi-instance segmentation scenario:

$$WID(\mathcal{P}, \mathcal{G}) = \sum_i^n \alpha(g_i)(1 - D(p_i, g_i)) + \sum_i^n \sum_{j \neq i}^n D(p_i, p_j) \tag{1}$$

where $\mathcal{P} = \{p_0, p_1 ..., p_n\}$ is the set of instance predictions and $\mathcal{G} = \{g_0, g_1 ..., g_n\}$ the set of instance ground-truth. D is the standard Dice coefficient, hence $(1 - D(p_i, g_i))$ corresponds to a non-weighted Dice loss for a single instance. $\alpha(g_i)$ is an instance specific weight coefficient that increases for smaller instances. We define $\alpha(g_i) = 1 - \frac{|g_i|}{WH}$, where $|g_i|$ is the total number of pixels in g_i and (W, H) correspond to the width and height of the video frame. Finally, $\sum_i^n \sum_{j \neq i}^n D(p_i, p_j)$ enforces a penalty for overlapping instance predictions reducing incorrect instance assignments.

Formally, WID is supported by the bijection between the Dice coefficient and the Jaccard index ($J = \frac{D}{2-D}$), which guarantees that minimizing a Dice loss function maximizes the associated Jaccard metric. In contrast to the Dice coefficient proposed by [37], whose convergence rate is modulated by $\frac{2}{(2-D)^2}$, WID presents an improvement as it considers instance overlapping errors and allows for an instance-specific weighting instead of the standard practice of class-specific weighting.

3.2. Attention Priors

At the core of our approach is a set of attention cues $A_{m,t}$ estimated for all m instances at time t given their visual appearance and location at time $t - n$. Initially, we estimate long-term dependencies (*i.e.* $n >> 1$), by means of a tracking strategy that creates m attention regions according to the temporal evolution of n previous positive detections. We call these cues Long-term Attention (LTA). We complement these long term recurrences with attention cues over short-term intervals (*i.e.* $n = 1$). We estimate a robust version of Optical Flow and use it as a fine-grained (pixel-level) short-term motion attention cue. We also propagate the prediction at time $t-1$ by warping it with the estimated optical flow, creating a coarser (region-level) Short-term Attention (STA). Figure 3 presents an overview of the selected cues.

Short-Term Attention Prior. We propose an STA source by including information from the warped segmentation mask of the previous frame. This prior is motivated by the video's causal nature, in which the predicted mask at time t is strongly related to the prediction at time $t - 1$. Such temporal correspondence approximates both the target specification and its immediate trajectory.

We follow the standard definition of a recurrent warping $w(S_t, \mathbf{o}_t) = S_{t+1}$ as the mapping $w : \mathbb{R}^{W \times H \times C} \times \mathbb{R}^{W \times H \times 2} \rightarrow \mathbb{R}^{W \times H \times C}$ over a frame S_t from a video V with width W, height H, number of channels C, and estimated optical flow field $\mathbf{o}_t$ between frames S_{t+1} and S_t. This short-term prior allows the network to explicitly assess
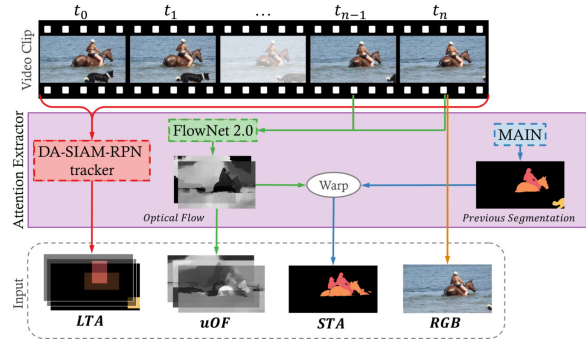

Figure 3. **Attention cues.** MAIN integrates multiple attention cues to create temporally consistent instance-level segmentations. During a forward pass of frame t_n, we calculate the Long-term Attention (LTA) cue, as a stack of bounding boxes of the target objects, calculated using DA-SIAM-RPN tracker; the Unit Optical Flow (uOF) using FlowNet2.0; and the Short-term Attention (STA) cue with the segmentation at time t_{n-1} produced by MAIN warped by the corresponding optical flow.

the approximate appearance of the targets resulting in more precise segmentations.

Long-Term Attention Prior. We establish an LTA prior from the iterative estimation of an instance location. We calculate this prior by means of a bounding box tracker, which estimates an approximate location of an object at frame t given its appearance and its immediate context at times $\{t - 1, ..., t - n\}$. This process is performed in a sequential manner, starting at frame 0, whose bounding box location information is known.

Since tracking is, in essence, a one-shot learning problem [2] and its core algorithm can be initialized and efficiently executed, even in large-scale datasets, it directly fits into the one-shot video instance segmentation framework.

Unit Optical Flow. Many recent methods for video object segmentation [50, 48, 47, 39] focus on inferring pixel-wise mappings exclusively from spatial data. Optionally, these methods complement their predictions with temporal estimates over initial ground-truth labels. We depart from this mainstream approach and estimate explicit temporal cues from the video, which we fuse at frame-level with the spatial information.

We compute the optical flow from consecutive RGB frames using FlowNet2.0 [21] and map the estimated vector field $\mathbf{o} = (x, y)$ into its unitary direction field $\hat{\mathbf{o}} = \frac{o}{|o|}$. We concatenate these unitary vectors with the normalized magnitude of the flow field and name this 3-channel cue the *Unit Optical Flow* (uOF). Compared to the raw vector field, the uOF is bounded between $[-1, 1]$, thus it is more stable in the presence of large displacements and estimation errors. We use the uOF as a complementary source of information in MAIN, thereby achieving end-to-end training using both spatial and temporal cues.

3.3. Multi-Scale Separable Decoder

Since richer contextual information aids at better modeling local information and overall segmentation score, recent works [12, 34, 56, 6, 42] establish connections between coarse and fine information to produce better segmentations. Our decoder uses multi-scale separable convolutional operators to augment its field-of-view. There are two elements at the core of this enhancement: dilated convolutions [55] that enables the exponential expansion of a neuron's receptive field without losing resolution [55, 6]; and separable convolutions that aim at factoring cross-channel and spatial correlations [9], reducing computation, increasing time efficiency and which we find to be beneficial for instance segmentation.

3.4. Implementation Details

We train MAIN using the Pytorch library [38]. We choose the ADAM optimizer [25] and train until convergence with an initial learning rate of 1×10^{-4} and learning rate annealing $\gamma = 0.1$ every 45000 iterations. The decoder layers are randomly initialized using the Xavier weight initialization [14]. To speedup training in the large-scale Youtube-VOS, which contains mostly frames of HD-quality at 1280×768, we rescale the training data to 256×416, leading to a batch-size of 23 in training using a Titan X GPU.

Multi-Scale Separable Decoder. For our encoder backbone, we use Resnet50 pretrained on Image-Net [10]. We drop the final global pooling and fully connected layers and augment the backbone with side outputs [15, 35] at the end of each block (the final layer before the pooling operator). We adopt the architectural pattern from Feature Pyramid Networks [32], in which feature maps before every pooling operation are independently up-sampled by a factor of 2 and post-processed with a 4-layer feature pooling stack. The first layer of the stack is a standard 1x1 separable convolution and ReLU non-linearity [27]. The remaining 3 layers are 3×3 separable convolutions with dilation factors of 1, 2 and 3 [55].

The up-sample step is performed with a bilinear interpolation followed by a 3×3 convolutional layer. The feature pooling and up-sampling process are performed independently over all the side-outputs of the encoder, thus, allowing incremental fusion (by an element-wise addition) of every up-sampled response map with the pooled feature map from the previous level.

4. Experimental Validation

In this section, we provide empirical evidence to support our approach. We proceed by validating each individual contribution and related design choice.

4.1. Datasets and Evaluation Metrics

Youtube-VOS [53]. It is the first large-scale video segmentation dataset consisting of 4453 high quality hand labeled videos, split across training (3471), validation (474) and test (508) sets. Following the one-shot formulation, this dataset provides the ground-truth segmentation for the first frame in any set. Xu. *et al.* [53] reserve the test set for the annual challenge, and designate the validation set, whose ground truth labels are also withheld, for algorithm evaluation and ranking over a dedicated evaluation server.

Since annotations in the Youtube-VOS validation set are withheld [53], we create a random split from the original training set. As a result, we obtain two subsets: *Train66* (2314 videos) and *Train33* (1157 videos) with known labels and use these sets to empirically validate each of our contributions.

To assess the final performance of our MAIN method and to compare our results against the state-of-the-art, we follow the standard evaluation methodology of the Youtube-VOS benchmark introduced in [53]. The two main evaluation metrics are: region similarity or Jaccard index ($\mathcal{J}$) and F-measure ($\mathcal{F}$) for contour accuracy and temporal stability. The Youtube-VOS validation set includes instances from categories not available in the training set, thus, creating two dataset-specific metrics for *seen* and *unseen* object categories.

DAVIS-17 [41]. We complement our empirical validation by assessing the effectiveness of MAIN on the DAVIS-17 dataset, which consists of 60 high quality videos for training and 30 for validation. This dataset is densely annotated but is two orders of magnitude smaller than Youtube-VOS. DAVIS-17 also evaluates $\mathcal{J}$ and $\mathcal{F}$ but without the distinction between seen and unseen categories.

4.2. Architectural Ablation Experiments

We now empirically assess our architectural design choices namely: dilated and separable convolutions in the decoder of our network. Table 1 summarizes the results. We train four versions of our network with different combinations (absence/presence) of separable and dilated convolutions in *Train66*. Evaluation in *Train33* shows that using separable and dilated convolutions is beneficial for our method.

Loss Function: We evaluate the suitability of the proposed WID as loss function by comparing it against other loss functions. This comparative study is summarized in Table 2. We compare against Dice Coefficient and Binary Cross Entropy by training three versions of our network in *Train66* with the different loss functions. WID clearly outperforms the other losses when evaluated in the *Train33* set.

Separable Convolutions	Dilated Convolutions	$\mathcal{J}$ Seen
χ	χ	0.603
χ	✓	0.606
✓	χ	0.613
✓	✓	**0.645**

Table 1. **Decoder architecture.** Comparison configurations of MAIN. We assess the performance of the different combinations of separable and dilated convolutions. Results presented on Youtube-VOS *Train33* set. Both Separable and Dilated Convolutions contribute to the performance of the method.

Loss Function	$\mathcal{J}$ Seen
Binary Cross Entropy	0.538
DICE	0.611
Weighted Instance DICE (WID)	**0.645**

Table 2. **Loss functions.** Comparison of loss functions. We train MAIN with different loss functions and test them on Youtube-VOS *Train33* set. The proposed WID outperforms other loss function.

4.3. Attention Priors

To evaluate the effect of the different attention cues on MAIN, we ablate each attention cue to show their importance separately and jointly. Table 3 summarizes the ablation experiments. We train eight different configurations of our network in *Train66* by incrementally adding each of the different attention cues. Each attention cue contributes to the performance of the network, with Long-term Attention being especially useful at improving MAIN's performance, and Short-term Attention more suitable for refining results.

Unit Optical Flow. We integrate the uOF by concatenating it with the RGB channels in the input. We initialize the first layer weights that correspond to the uOF inputs as the RGB average weights, thus adding 3 channels to the input tensor. During training, we switch between the forward and backward estimates of the uOF. To verify the effectiveness of uOF compared to the raw optical flow, we train two versions of MAIN in *Train66* switching this attention cue and

Attention	Input Data	$\mathcal{J}$ Seen
None	RGB	0.321
None	RGB+uOF	0.326
STA	RGB	0.436
STA	RGB+uOF	0.500
LTA	RGB	0.625
LTA	RGB+uOF	0.628
LTA+STA	RGB	0.632
LTA+STA	RGB+uOF	**0.645**

Table 3. **Attention cues.** Comparison of long term vs short term attention priors on Youtube-VOS *Train33* set. The proposed long term and short term cues enable the multi-instance segmentation task, and complement each other when used in conjunction.

Optical Flow Configuration	$\mathcal{J}$ Seen
Optical Flow	0.612
Unit Optical Flow	**0.645**

Table 4. **Robust optical flow.** Comparison of optical flow sources used as input for video segmentation. The Unit Optical Flow has a better performance compare to using the raw Optical Flow field.

evaluate them in *Train33* . Table 4 shows that using uOF is more beneficial to MAIN than using raw optical flow. We also test the performance of the model when adding the uOF. Table 3 shows that for every configuration of attention cues the method benefits with the inclusion of uOF.

Multi-Instance Attention. We complement the input of the multi-scale encoder-decoder with a set of $2N$ attention maps (N for LTA and N for STA), in which N is set to the maximum number of instances in the dataset (6 for Youtube-VOS and 10 for DAVIS-17). Each attention map encodes the estimated location of a single instance by means of a binary bounding box. For a video with M instances ($M < N$), we set the remaining $N - M$ maps to 0. We concatenate these additional $2N$ channels to the RGB and uOF input. Since this modification changes the input dimension, we initialize the first layer weights corresponding to the attention cues with the average of the original RGB weights, thus, avoiding the need to retrain the whole layer and favoring faster convergence.

We evaluate the effectiveness of producing a multi-instance segmentation compared to estimating multiple single-instance predictions and then joining the results. Table 5 shows that the multi-instance approach of MAIN significantly outperforms the single-instance scenario.

Number of instances	$\mathcal{J}$ Seen
Single-Instance	0.616
Multi-Instance	**0.645**

Table 5. **Multi-instance predictions.** Comparison of multi-instance vs single-instance configuration on Youtube-VOS *Train33* set for MAIN. Multi-instance achieves a significantly better performance than single-instance.

Long Term Attention Priors. We derive our Long-term Attention from the tubes generated by the Distractor-Aware Siamese Tracker (DA-Siam-RPN) [57]. We keep the default anchor-ratios from the VOT-2018 dataset [26] and set the number of candidate scales to 8, with a displacement penalty of 0.055 windowed by a cosine function, and template learning rate of 0.395. For the network weights, we use the 'AlexNet-Big' model. The tracker is initialized with a tight bounding box created over the instance annotation from the first frame and run over the full frame-set of Youtube-VOS. In the **supplementary material**, we show a comparison of the relative performance of the proposed attention methods according to the tracker's average overlap

as the video progresses.

Long-term Attention is one of the most important cues for MAIN. Table 3 shows the improvement of adding the LTA for each of MAIN configuration. We find it beneficial to initially train MAIN using the ground-truth segmentation to create perfect bounding boxes and later, when the network converges, introduce the estimated LTA. This way, we allow the network to first learn to operate over attention cues and later to learn from error modes on the tracker.

Short Term Attention Priors. During the final training stage, we concatenate the STA attention to our input tensor. We extend the first layer weights, corresponding to the STA, by replicating the weights of the LTA input. Therefore, the final version of MAIN, has an input tensor of dimension $(6 + 2N) \times W \times H$ for the inputs: RGB, uOF, LTA, and STA. To train this last phase, we perform several data augmentation at different image scales and crops of size 256×416. We replace the ideal LTA cue with the one estimated by the tracker, and for the STA at time t, we randomly choose the ground-truth segmentation from consecutive frames $[t-1, t+1]$. We randomly dilate and erode the selected annotation with squared kernels of sizes that vary between 6 to 30 pixels, perform affine transformations such as scale change between 0.8 and 1.2 of ratio, and shift between 0 to 1% of image size. All these transformations are used to approximate the possible errors that the previous segmentations might have in the validation set. In the forward phase, we simply take the previously segmented output and warp it to the current time position by using the backward estimated optical flow.

To adapt MAIN to the error pattern in the tracker, we use a Curriculum Learning [1] strategy. We steadily increase the error source (*i.e.* tracker error) as the optimization process advances. We start by fine-tuning MAIN with LTA and STA data over the first 2, 4, 8 and 14 frames for three epochs each. Table 3 shows the significant improvement of adding the STA for each MAIN configuration.

Method	OnT	$\mathcal{J}$ seen	$\mathcal{J}$ unseen	$\mathcal{F}$ seen	$\mathcal{F}$ unseen
S2S [53]	✓	0.710	0.555	0.700	0.612
MAIN	χ	**0.667**	**0.550**	**0.690**	**0.630**
S2S [53]	χ	0.667	0.482	0.655	0.503
OnAVOS [48]	✓	0.601	0.466	0.627	0.514
OSMN [54]	✓	0.600	0.406	0.601	0.440
MaskTrack [39]	✓	0.599	0.450	0.595	0.479
OSVOS [5]	✓	0.598	0.542	0.605	0.607

Table 6. **Comparison of State-of-the-art methods** We evaluate MAIN on Youtube-VOS validation set, scores are taken from [53]. For each metric we highlight in red the best result and in blue the second best result. Results in **bold** correspond to our method. OnT is the abbreviation of Online Training.

Method	OnT	$\mathcal{J}$	$\mathcal{F}$
MAIN	χ	**0.602**	**0.657**
VideoMatch [20]	χ	0.565	-
FAVOS [7]	χ	0.546	0.618
OSMN [54]	χ	0.525	0.571
SiamMask [49]	χ	0.511	0.550
MaskRNN [19]	χ	0.455	-
DyeNet [30]	✓	0.673	0.710
OnAVOS [48]	✓	0.640	0.712
VideoMatch [20]	✓	0.614	-
OSMN [54]	✓	0.608	0.571
MaskRNN [19]	✓	0.605	-
OSVOS [5]	✓	0.566	0.639

Table 7. **DAVIS 17 benchmark.** Comparison of State-of-the-art methods on DAVIS Validation. Offline training methods are shown separated for a fair comparison. For each metric we highlight in red the best result and in blue the second best result. Results in **bold** correspond to our method.

5. Comparison with the State-of-the-art

In this section, we compare our best MAIN network (RGB+uOF+LTA+STA) against state-of-the-art methods. Table 6 summarizes this comparison for the task of instance video segmentation on the Youtube-VOS validation set. We follow the standard testing methodology proposed by [53]. We distinguish between offline and online methods for a fair comparison. As outlined in Section 4.1, the Youtube-VOS dataset breaks up the evaluation metrics between seen and unseen categories. This distinction creates a large performance gap between both sets, the latter being far more difficult.

MAIN achieves state-of-the-art scores for every metric in the offline configuration, achieving an accuracy that is competitive compared to online methods. Our method relies on generic attention cues and favors consistent temporal predictions along the video sequence, instead of enforcing strong semantics or individual appearance models. This leads to a significant performance improvement in the un-
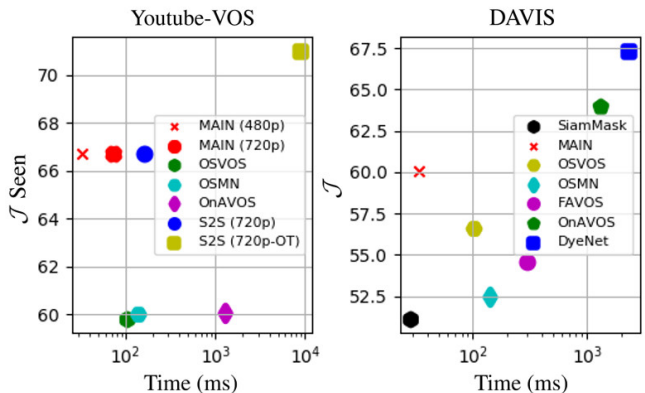

Figure 4. **Efficiency Benchmark.** Jaccard metric vs forward time (ms) on the Youtube-VOS and DAVIS 2017 datasets. Our method has a clear advantage on both benchmarks running on near real-time on DAVIS 2017.

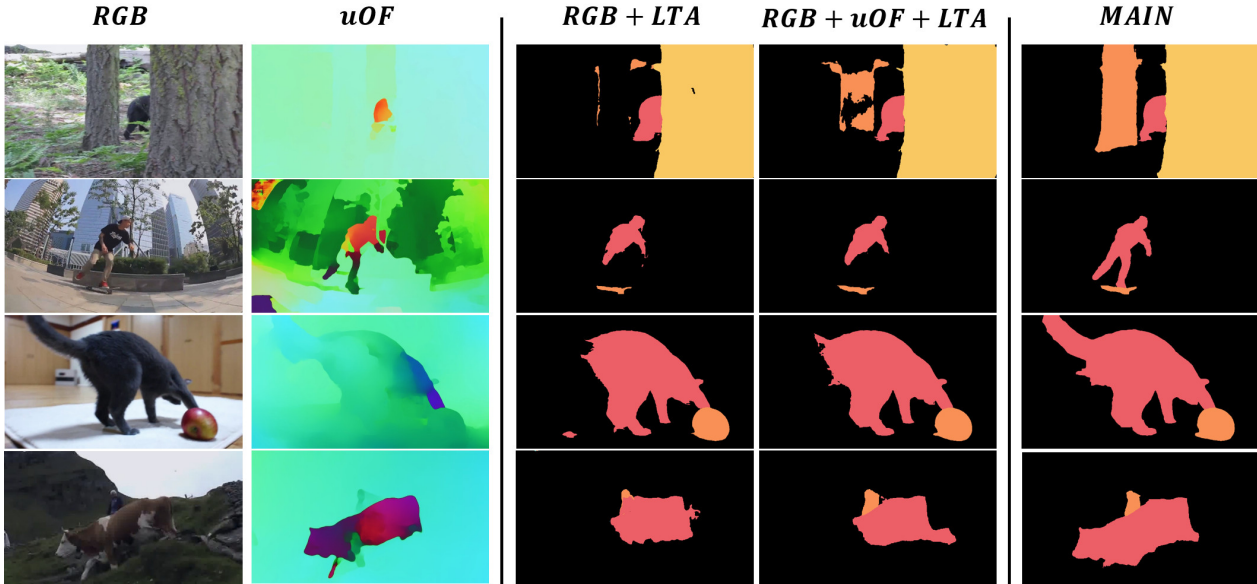| RGB | uOF | RGB + LTA | RGB + uOF + LTA | MAIN |

Figure 5. **Qualitative results.** Each row represents a frame from a specific video. From left to right, the first column represents the original image. The second column is a representation of the Unitary Optical Flow. The third and fourth columns correspond to the results using only the configuration of $RGB + LTA$ and $RGB + uOF + LTA$ correspondingly. The last column corresponds to MAIN results.

seen scenario, in which MAIN surpasses the current offline state-of-the-art by 6.8% and 12.7% in $\mathcal{J}$ and $\mathcal{F}$ metrics, respectively. Compared against the online methods, MAIN results are higher than almost all methods except for [53], where MAIN is within 0.5% of the latter's performance in $\mathcal{J}$ unseen but outperforms it by 1.8% in $\mathcal{F}$.

We also evaluate MAIN against the state-of-the-art on the DAVIS validation dataset, summarized in Table 7. We achieve state-of-the-art performance on the offline task, while remaining competitive against online methods. Furthermore, MAIN fares favorably against approaches that also rely on spatial attention like [54], and long-term recurrences like [49, 19].

5.1. Efficiency Analysis

We conclude this section with an efficiency analysis. Figure 4 shows the comparison between performance and inference time. We calculate the inference time for a single frame in MAIN after estimating the attention mechanisms. MAIN outperforms most state-of-the-art methods by an order of magnitude, where only SiamMask [49] performs faster in inference. It is important to emphasize that SiamMask was devised for fast inference and underperforms in $\mathcal{J}$ and $\mathcal{F}$ when compared to MAIN. Additionally, SiamMask can only generate a single instance segmentation per forward pass. Hence, MAIN is faster than SiamMask in sequences with more than one instance.

6. Qualitative Results

To complement our experimental validation, we present some qualitative results in Figure 5 that show the visual differences of incrementally adding cues in training. On the first (top) row, MAIN overcomes a challenging scenario with occlusions, similar visual instances, and blur. The second row corresponds to a video with fast moving objects, different-sized objects, and complex motion patterns. In this case, motion and STA cues play a key role in the good performance because all instances tend to stay in the center, while the background is moving quickly. The third video shows a case with objects with distinct degrees of blur and with an incomplete label in the first frame (the tail of the cat is missing). Even though MAIN is not aware that cats have tails, it deduces it thanks to the attention mechanisms that ensure that the tail is part of the object. The final video (bottom) shows two segmentation targets one mostly occluded an the other completely visible and in motion. While the LTA and uOF approximate the location and number of instances, only the inclusion of STA approximates the targets shape.

Overall, there is an improvement when stacking different priors and sources of information. LTA mostly approximates the objects location but fails at providing a good recall, especially for finer details or highly textured objects. Using the uOF refines the prediction of the objects with somewhat homogeneous motion. Finally, incorporating STA with LTA and uOF enhances the segmentation quality by reducing the false negatives and improving the boundary definition between adjacent masks. We include more examples in the **supplemental material** that show MAIN in diverse scenarios.

7. Conclusions

MAIN is an efficient approach devised for generic instance-level segmentation. It is trained end-to-end with a new WID loss suitable for class imbalance, generating multiple instance-level segmentations in a single forward pass. We validate MAIN in the first large-scale video segmentation benchmark, achieving state-of-the-art results while running at 30.3 FPS. The increments in the unseen metrics demonstrate MAIN's effectiveness at fusing generic grouping cues and producing temporally consistent segmentations without requiring domain-specific knowledge.

References

[1] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009. 7

[2] L. Bertinetto, J. F. Henriques, J. Valmadre, P. Torr, and A. Vedaldi. Learning feed-forward one-shot learners. In *NeurIPS*, 2016. 4

[3] G. J. Brostow, J. Fauqueur, and R. Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 2009. 2

[4] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, 2010. 3

[5] 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. 1, 2, 7

[6] 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*, 2018. 5

[7] J. Cheng, Y.-H. Tsai, W.-C. Hung, S. Wang, and M.-H. Yang. Fast and accurate online video object segmentation via tracking parts. In *CVPR*, 2018. 1, 3, 7

[8] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang. Segflow: Joint learning for video object segmentation and optical flow. In *ICCV*, 2017. 3

[9] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, 2017. 5

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

[11] R. Deng, C. Shen, S. Liu, H. Wang, and X. Liu. Learning to predict crisp boundaries. In *ECCV*, 2018. 3

[12] Y. Du, W. Wang, and L. Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *CVPR*, 2015. 5

[13] F. Galasso, N. S. Nagaraja, T. J. Cárdenas, T. Brox, and B. Schiele. A unified video segmentation benchmark: Annotation, metrics and analysis. In *ICCV*, 2013. 2

[14] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010. 5

[15] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, 2015. 5

[16] G. E. Hinton. Connectionist learning procedures. *Artificial intelligence*, 1989. 2

[17] S. Hochreiter and J. Schmidhuber. Lstm can solve hard long time lag problems. In *NeurIPS*, 1997. 2

[18] P. Hu, G. Wang, X. Kong, J. Kuen, and Y.-P. Tan. Motion-guided cascaded refinement network for video object segmentation. In *CVPR*, 2018. 3

[19] Y.-T. Hu, J.-B. Huang, and A. Schwing. Maskrnn: Instance level video object segmentation. In *NeurIPS*, 2017. 1, 3, 7, 8

[20] Y.-T. Hu, J.-B. Huang, and A. G. Schwing. Videomatch: Matching based video object segmentation. In *ECCV*, 2018. 7

[21] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. *arXiv preprint arXiv:1612.01925*, 2016. 4

[22] V. Jampani, R. Gadde, and P. V. Gehler. Video propagation networks. In *CVPR*, 2017. 3

[23] M. Keuper, B. Andres, and T. Brox. Motion trajectory segmentation via minimum cost multicuts. In *ICCV*, 2015. 3

[24] A. Khoreva, R. Benenson, E. Ilg, T. Brox, and B. Schiele. Lucid data dreaming for multiple object tracking. *arXiv preprint arXiv:1703.09554*, 2017. 1, 2, 3

[25] D. Kinga and J. B. Adam. A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 5

[26] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. C. Zajc, T. Vojir, G. Bhat, A. Lukezic, A. Eldesokey, G. Fernandez, and et al. The sixth visual object tracking vot2018 challenge results, 2018. 6

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 2, 5

[28] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1989. 2

[29] S. Li, B. Seybold, A. Vorobyov, X. Lei, and C.-C. Jay Kuo. Unsupervised video object segmentation with motion-based bilateral networks. In *ECCV*, 2018. 3

[30] X. Li and C. Change Loy. Video object segmentation with joint re-identification and attention-aware mask propagation. In *ECCV*, 2018. 1, 7

[31] X. Li and C. C. Loy. Video object segmentation with joint re-identification and attention-aware mask propagation. *arXiv preprint arXiv:1803.04242*, 2018. 1

[32] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 5

[33] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 3

[34] W. Liu, A. Rabinovich, and A. C. Berg. Parsenet: Looking wider to see better. *arXiv preprint arXiv:1506.04579*, 2015. 5

[35] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 2, 5

[36] K.-K. Maninis, J. Pont-Tuset, P. Arbeláez, and L. Van Gool. Convolutional oriented boundaries. In *ECCV*, 2016. 3

[37] F. Milletari, N. Navab, and S.-A. Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *Fourth International Conference on 3D Vision*. IEEE, 2016. 3, 4

[38] 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. In *NeurIPS-Workshop*, 2017. 5

[39] F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung. Learning video object segmentation from static images. In *CVPR*, 2017. 2, 3, 4, 7

[40] 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 *CVPR*, 2016. 2

[41] 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 preprint arXiv:1704.00675*, 2017. 2, 3, 5

[42] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015. 5

[43] A. Shaban, A. Firl, A. Humayun, J. Yuan, X. Wang, P. Lei, N. Dhanda, B. Boots, J. M. Rehg, and F. Li. Multiple-instance video segmentation with sequence-specific object proposals. In *CVPR Workshop*, 2017. 1

[44] J. Shin Yoon, F. Rameau, J. Kim, S. Lee, S. Shin, and I. So Kweon. Pixel-level matching for video object segmentation using convolutional neural networks. In *ICCV*, 2017. 1

[45] P. Sundberg, T. Brox, M. Maire, P. Arbeláez, and J. Malik. Occlusion boundary detection and figure/ground assignment from optical flow. In *CVPR*, 2011. 2

[46] P. Tokmakov, K. Alahari, and C. Schmid. Weakly-supervised semantic segmentation using motion cues. In *ECCV*, 2016. 2, 3

[47] P. Tokmakov, K. Alahari, and C. Schmid. Learning motion patterns in videos. In *CVPR*, 2017. 1, 3, 4

[48] P. Voigtlaender and B. Leibe. Online adaptation of convolutional neural networks for the 2017 davis challenge on video object segmentation. In *The 2017 DAVIS Challenge on Video Object Segmentation-CVPR Workshops*, 2017. 1, 2, 4, 7

[49] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. Torr. Fast online object tracking and segmentation: A unifying approach. *arXiv preprint arXiv:1812.05050*, 2018. 7, 8

[50] W. Wang and S. Bing. Super-trajectory for video segmentation. *arXiv preprint arXiv:1702.08634*, 2017. 3, 4

[51] H. Xiao, J. Feng, G. Lin, Y. Liu, and M. Zhang. Monet: Deep motion exploitation for video object segmentation. In *CVPR*, 2018. 2, 3

[52] S. Xie and Z. Tu. Holistically-nested edge detection. In *ICCV*, 2015. 3

[53] N. Xu, L. Yang, Y. Fan, D. Yue, Y. Liang, J. Yang, and T. Huang. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018. 2, 3, 5, 7, 8

[54] L. Yang, Y. Wang, X. Xiong, J. Yang, and A. K. Katsaggelos. Efficient video object segmentation via network modulation. *Algorithms*, 2018. 1, 2, 7, 8

[55] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 5

[56] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *CVPR*, 2017. 5

[57] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu. Distractor-aware siamese networks for visual object tracking. In *ECCV*, 2018. 6

8. Appendix

8.1. Temporal Consistency Analysis

To complement the experimental validation of section 4, we evaluate the temporal consistency of our method. Figure 6 shows the performance of the different versions of our method (different choices for attention cues) as the video sequences progress. For this analysis we train on *Train66* and validate over the *Train33* set.

Our method incrementally benefits with the addition of each attention prior. LTA is critical for MAIN's performance, if MAIN only considers STA the scores drop much faster during the first frames. We explain this behaviour as errors propagate faster if STA is the only source of attention. Overall, LTA improves the temporal consistency of predictions, enabling mask refinement by STA. Finally the combination of RGB, uOF and STA has a slower decrease than adding RGB+STA, which shows a complementary behaviour of the pixel-level and region-level short-term attention cues.
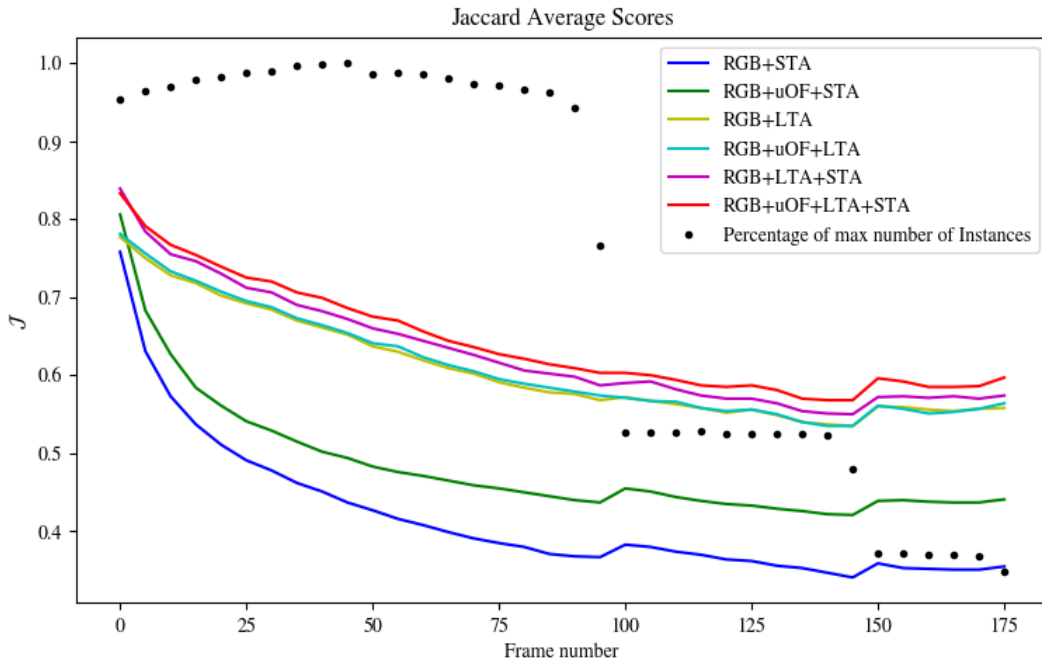

Figure 6. **Temporal Consistency.** Mean Jaccard scores along time according to the selected attention methods in the YoutubeVOS *Train33* set.

Like most video object segmentation methods, the performance of MAIN slowly decreases as the sequence progresses, it seems to stabilize around frame 100, above the 0.6 mean IoU. We will also note that the graph presents an unusual jump in performance around frames 100 and 150, specially for those configurations that only consider short-term priors. This particular behaviour is due to a bias in the videos length. In figure 6 the black dots represents the number of instances still present at frame n as fraction of the maximum number instances present at any time, the sharp declines in the number of instances correlate with the sudden jumps in the $\mathcal{J}$ of our methods, it seems that the remaining group of videos at frames 100 and 150 contain instances or scenarios that are easier to segment.

8.2. Qualitative Results

We provide more qualitative examples to better understand the capabilities and failure modes of MAIN. Figures 7 to 18 show challenging scenarios that include: occlusions (figures 7, 8, 11, 12, 17 and 18), scale changes (figures 10, 9, 14), appearance changes (figures 8, 13, 9 and 15), multiple similar instances (figures 12, 13 and 13), fast motion (figures 10, 8, 9 and 17) and very target small objects (10, 15, 16).

Figure 7. Exhibits three subjects and their Skiing equipment. This video provides a challenging scenario for MAIN due to the small size of the instances (see the snowboard in yellow just below the person in orange). Additionally the person in green gets completely occluded during the sequence. MAIN is robust to false positive detections of the person in green, while still being accurate at segmenting the other small instances on the scene. To visualize the animated figure use Adobe Acrobat Reader.

Figure 8. Contains a difficult scenario for methods like MAIN that do not directly rely on semantics, here there is a large amount of overlap between the girl and the dog, moreover the quick motion patterns of the dog are a large source of errors for all the attention cues. Nevertheless MAIN manages to create an almost accurate boundary around the girl and has a high recall for the pixels in the dog instance. Its main source of error are wrong instance assignments between the girl and the dog, specially the girl's arm which has a similar color to the dog's fur.

Figure 9. Shows an interesting scenario with a mirror that reflects the segmentation targets. There are minimal false positive detections located at the boundary between the small ape and its reflection. The main source of errors are erroneous instance label when the instances overlap.

Figure 10. Shows a complex scenario where attention cues must be initialized from small, fast moving objects, namely three bikes and riders with very similar visual features and small size. While our algorithm manages to identify the objects of interest, it fails when the fast moving objects become smaller and generates overlapping predictions with false negatives.

Figure 11. Illustrates an occlusion scenario that generates errors in the segmentation mostly due to the visual similarity of the segmentation target and its occluding background, along with the unusual lighting of this particular underwater scene. In this video MAIN propagates wrong estimations of the mask trough the sequence generating a fake segmentation mask around the occlusion boundary.

Figure 12. This is a hard scenario for MAIN, where visually similar instances overlap and occlude each other in complex patterns during a large period of time. While MAIN is still capable of identifying the target instances, it mixes their label information.

Figure 13. Presents an ideal segmentation scenario for MAIN, the instances appearance is almost unchanged trough the sequence, smooth motion patterns favor the propagation of attention cues and there is no overlap between semantically similar objects in the scene. These conditions largely favor the accurate propagation of attention priors. This video is penalized mostly by errors at segmenting fine-grain details like the sheep's legs and face boundaries.

Figure 14. This video shows a relative simple scene with an single large object of interest. However, the size of the instance changes drastically along the video given its fast motion. This affects the estimation of all the priors leading to false positives.
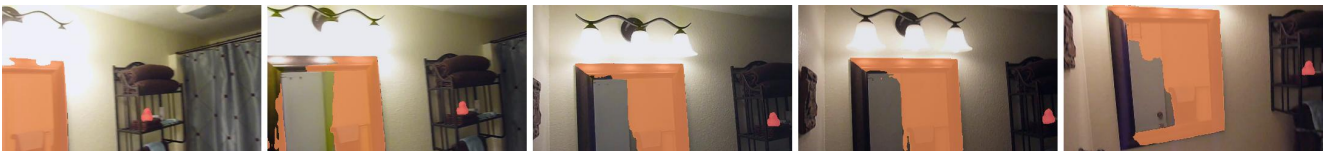

Figure 15. This video displays a scene with two objects of interest. The first one is a small object and the second one is a large mirror. Despite the small size of the first object, MAIN is capable to consistently segment it along the sequence. The mirror is a very hard segmentation target because of its constant change in 'appearance'. MAIN seems to attend to the reflected objects (a green door) and the mirror's frame.

Figure 16. This figure visualizes three small objects that interact with each other. In this particular setting, the optical flow doesn't provide mayor clues, as the motion of the surrounding plants and the apparent motion of the water results in much larger local motion estimations than those of the segmentation targets. However, MAIN achieves a great segmentation over all three objects.

Figure 17. This scene shows five instances. Two of them are partially occluded. Nonetheless, MAIN maintains a qualitatively acceptable segmentation until the last frames where it gets confused with the incoming truck.
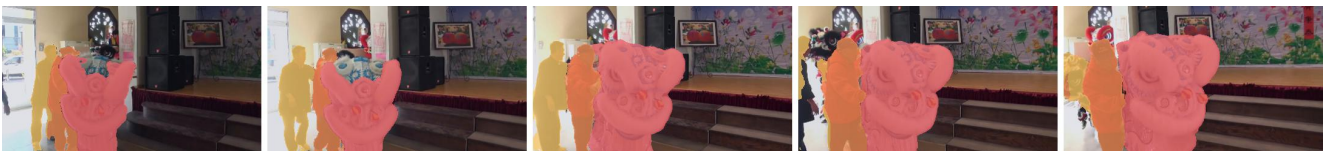

Figure 18. Evidences two very common instances (humans), and an extremely unusual dragon-like figure. In this scenario, all three instances are segmented until the person in yellow disappears. It is remarkable that MAIN provides a mostly accurate segmentations of a complex instance whose appearance and scale change over time, correcting some initial false negatives