Learning Video Object Segmentation with Visual Memory*

Presented by Emanuela Haller

Pavel Tokmakov, Karteek Alahari, Cordelia Schmid
Inria

Frame: 00001 | Ground Truth | LVO - J=0.758 - F=0.781

The task of segmenting moving objects in unconstrained videos

- Input
  - Video frames & estimated optical flow
- Output
  - Binary segmentations of moving objects
    - Moving objects = move in at least one frame

- Two-stream neural network

  1) encode spatial and temporal features

  2) capture the evolution of objects over time

Contributions

- The solution does not require manually annotated frames in the input video
- The network incorporates a memory unit to capture evolution of objects
- Exploit CNN representations instead of hand-crafted features
- Learn features vs propagation of initial guess

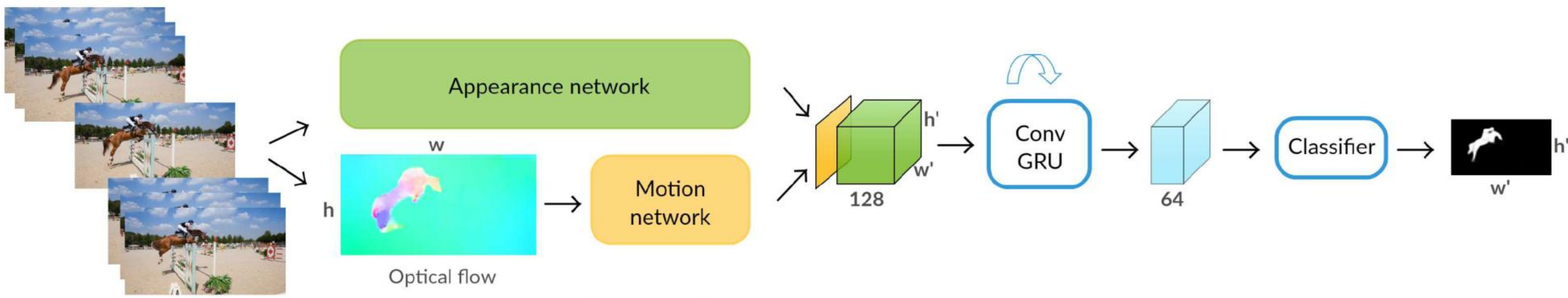
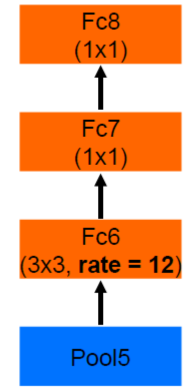


Figure 2. Overview of our segmentation approach. Each video frame is processed by the appearance (green) and the motion (yellow) networks to produce an intermediate two-stream representation. The ConvGRU module combines this with the learned visual memory to compute the final segmentation result. The width (w') and height (h') of the feature map and the output are $w/8$ and $h/8$ respectively.

Appearance network

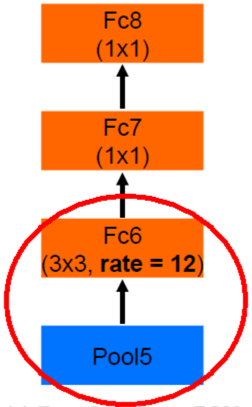


(a) DeepLab-LargeFOV

- DeepLab-LargeFOV*
  - Atrous convolution in VGG-16 'fc6' layer
    - Relatively high spatial resolution of features
    - Context information
- Pretrained on PASCAL VOC 2012 for semantic segmentation
  - Distinguish objects from background as well as from each other

"Semantic image segmentation with deep convolutional nets and fully connected CRFs"- ICLR 2015
L.C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille

Appearance network



(a) DeepLab-LargeFOV

- w'' x h'' x 1024
- Two 1x1 convolutional layers
  - Trained along with ConvGRU
- w' x h' x 128
  - w' = w/8, h' = h/8

Motion network

- MPNet*

- Pretrained on FlyingThings3D dataset
  - Synthetic dataset

- w/4 x h/4 x 1 motion prediction output
  - Likelihood of the corresponding pixel being in motion
  - Downsampled => w/8 x h/8 x 1

“Learning Motion Patterns in Videos”- CVPR 2017
Pavel Tokmakov, Karteek Alahari, Cordelia Schmid

Motion network

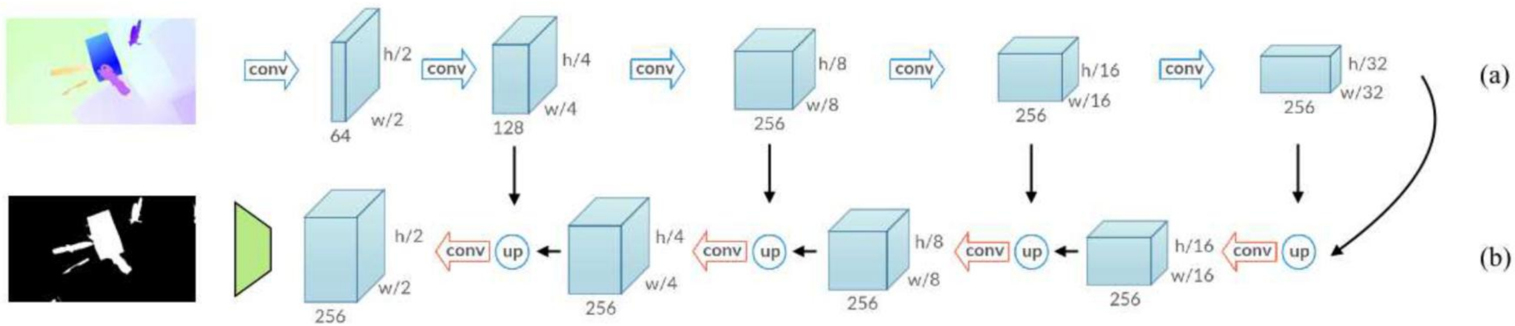


Figure 3. Our motion pattern network: MP-Net. The blue arrows in the encoder part (a) denote convolutional layers, together with ReLU and max-pooling layers. The red arrows in the decoder part (b) are convolutional layers with ReLU, 'up' denotes 2×2 upsampling of the output of the previous unit. The unit shown in green represents bilinear interpolation of the output of the last decoder unit.

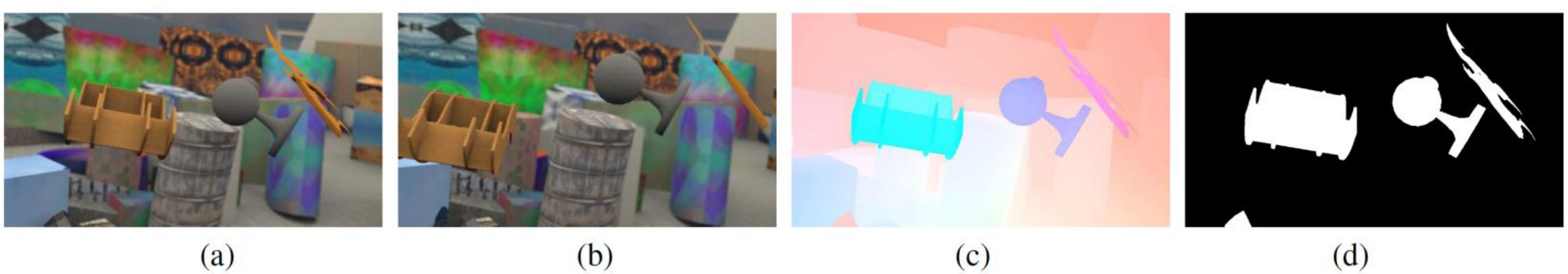


Figure 2. (a,b) Two example frames from a sequence in the FlyingThings3D dataset [23]. The camera is in motion in this scene, along with four independently moving objects. (c) Ground-truth optical flow of (a), which illustrates motion of both foreground objects and background with respect to the next frame (b). (d) Ground-truth segmentation of moving objects in this scene.

MPNet

Limitations:

- Frame based approach
- Overlooks appearance features
- Fails if the object stops moving (no motion cues)

Solutions:

- Heuristic post-processing step with object cues
- Combine with other video segmentation methods
- CRF

Memory module

- Based on convolutional gated units – ConvGRU
- Goal:
  - Refine estimates of appearance and motion networks
  - Memorize the appearance and location of objects
- Helps in frames where:
  - Objects are static
  - Motion prediction fails

Memory module

$$
\begin{aligned}
z_t &= \sigma(x_t * w_{xz} + h_{t-1} * w_{hz} + b_z), && (1)\\
r_t &= \sigma(x_t * w_{xr} + h_{t-1} * w_{hr} + b_r), && (2)\\
\tilde{h}_t &= \tanh(x_t * w_{x\tilde{h}} + r_t \odot h_{t-1} * w_{h\tilde{h}} + b_{\tilde{h}}), && (3)\\
h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t, && (4)
\end{aligned}
$$

$$
\begin{cases}
x_t \ \textit{two stream representation}\\
h_{t-1} \ \textit{previous state}\\
h_t \ \textit{current state}\\
z_t \ \textit{update gate}\\
r_t \ \textit{forget gate}\\
\tilde{h}_t \ \textit{candidate memory}
\end{cases}
$$

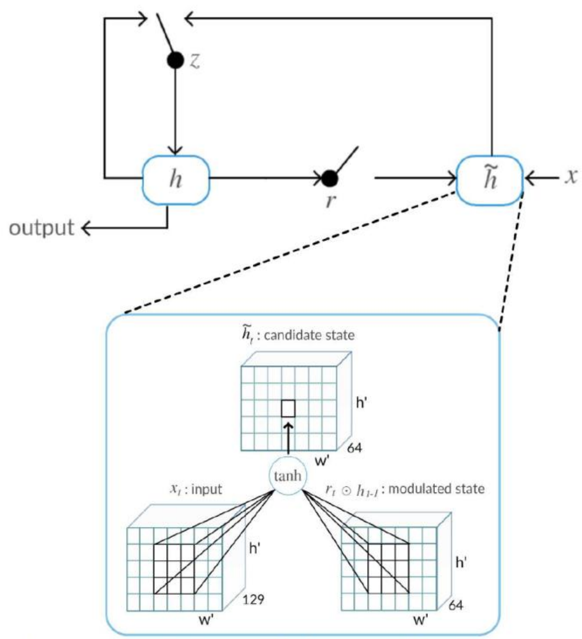


Figure 3. Illustration of ConvGRU with details for the candidate hidden state module, where $\tilde{h}_t$ is computed with two convolutional operations and a tanh nonlinearity.

Memory module

- Visual memory representation of a pixel is determined not only by the input and the previous state at that pixel, but also by its local neighborhood.

Memory module

- Bidirectional processing
  - Handle cases where objects move in the latter frames
  - Improves the ability to correct motion prediction errors

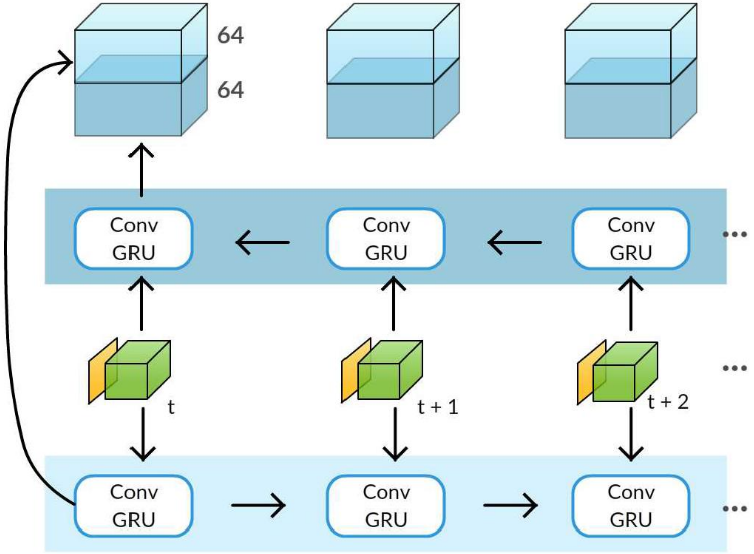


Figure 4. Illustration of the bidirectional processing with our ConvGRU module.

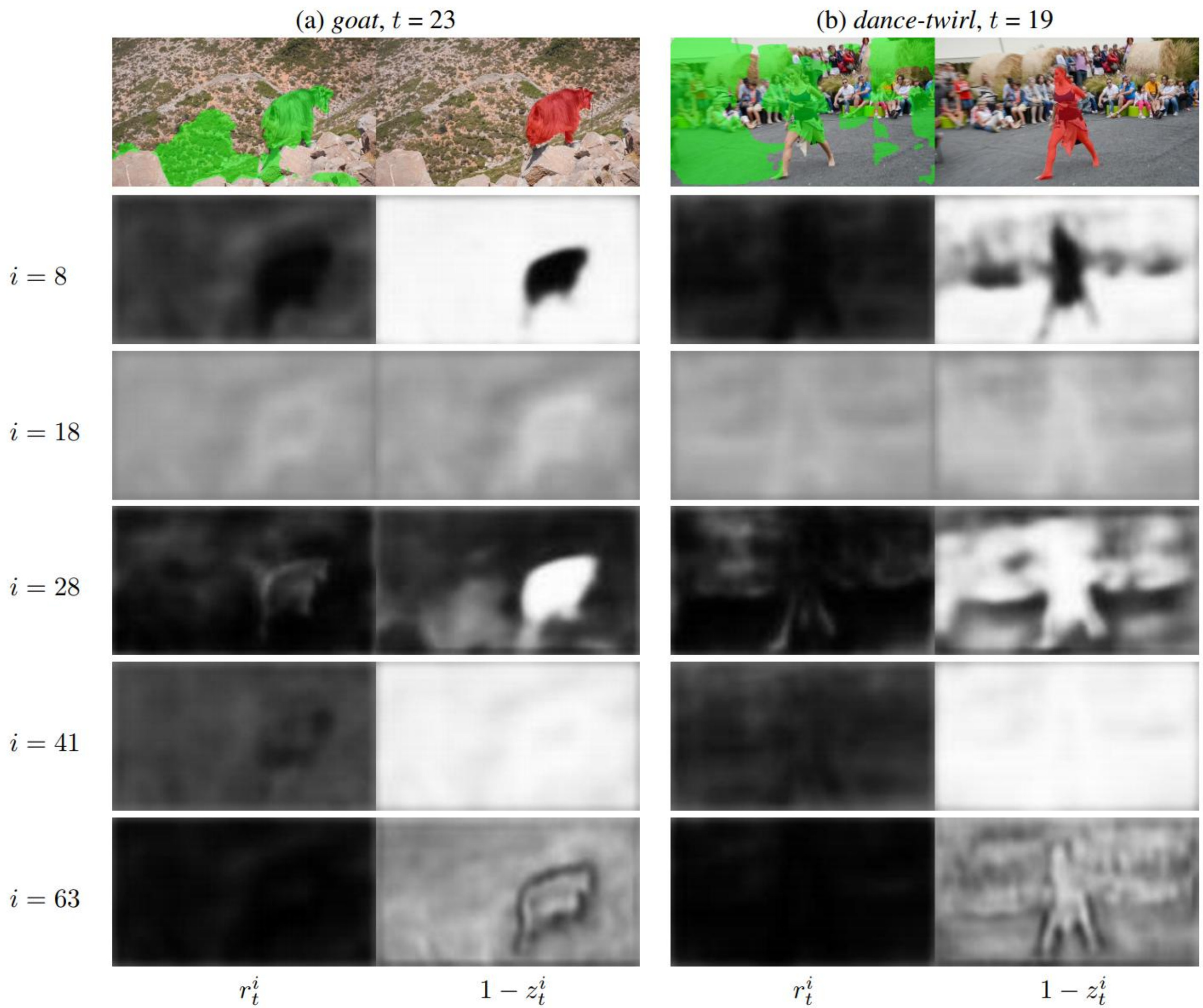


Figure 7. Visualization of the ConvGRU gate activations for two sequences from the DAVIS validation set. The first row in each example shows the motion stream output and the final segmentation result. The other rows are the reset (r_t) and the inverse of the update ($1 - z_t$) gate activations for the corresponding ith dimension. These activations are shown as grayscale heat maps, where white denotes a high activation.

Training

- Only ConvGRU

- DAVIS dataset
  - 30 videos

- Augmentation
  - Simulate stop-and-go scenarios

Ablation study

Aspect	Variant	Mean IoU
Ours (fc6, ConvGRU, Bidir, DAVIS)		70.1
App stream	no	43.5
	RGB	58.3
	2-layer CNN	60.9
	DeepLab fc7	69.8
	DeepLab conv5	67.7
App pretrain	ImageNet only	64.1
Motion stream	no	59.6
Memory module	ConvRNN	68.7
	ConvLSTM	68.9
	no	64.1
Bidir processing	no	67.2
Train data	FT3D GT Flow	55.3
	FT3D LDOF Flow	59.6

Table 1. Ablation study on the DAVIS validation set showing variants of appearance and motion streams and memory module. "Ours" refers to the model using fc6 features together with a motion stream, and a bidirectional ConvGRU trained on DAVIS.

Comparison to MPNet

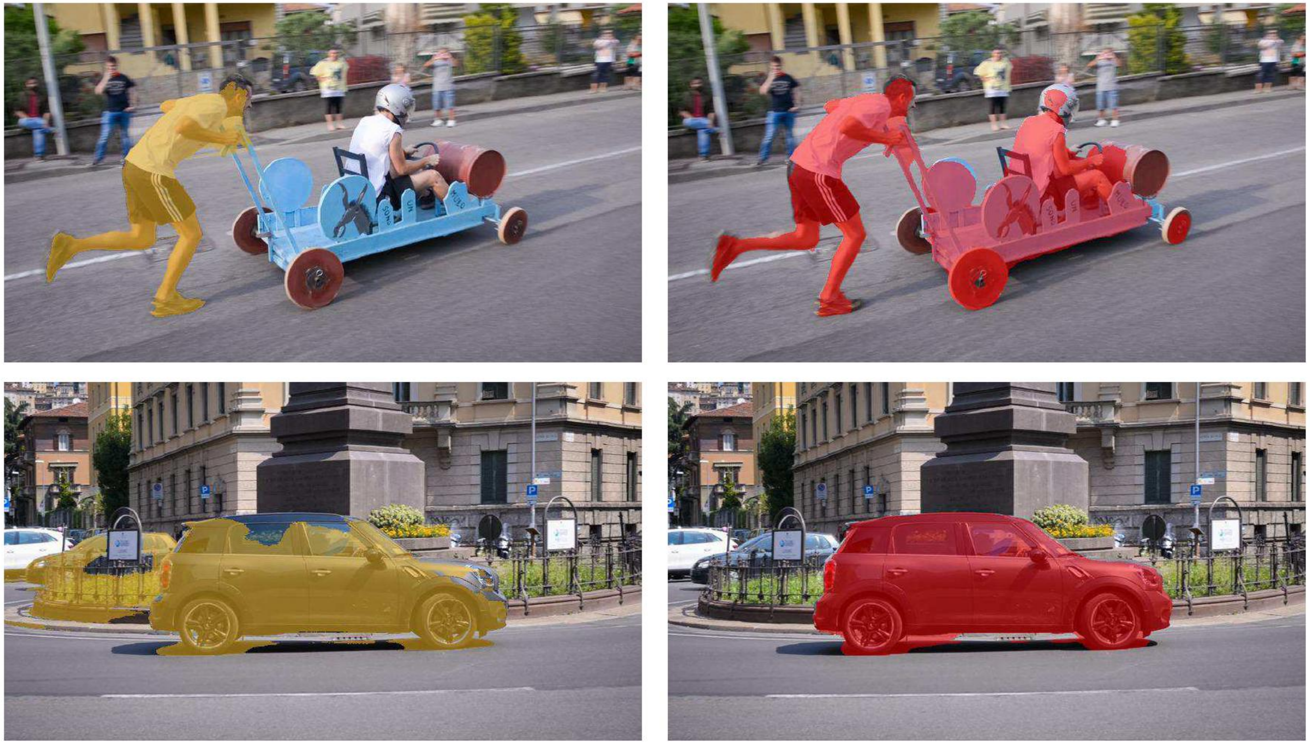


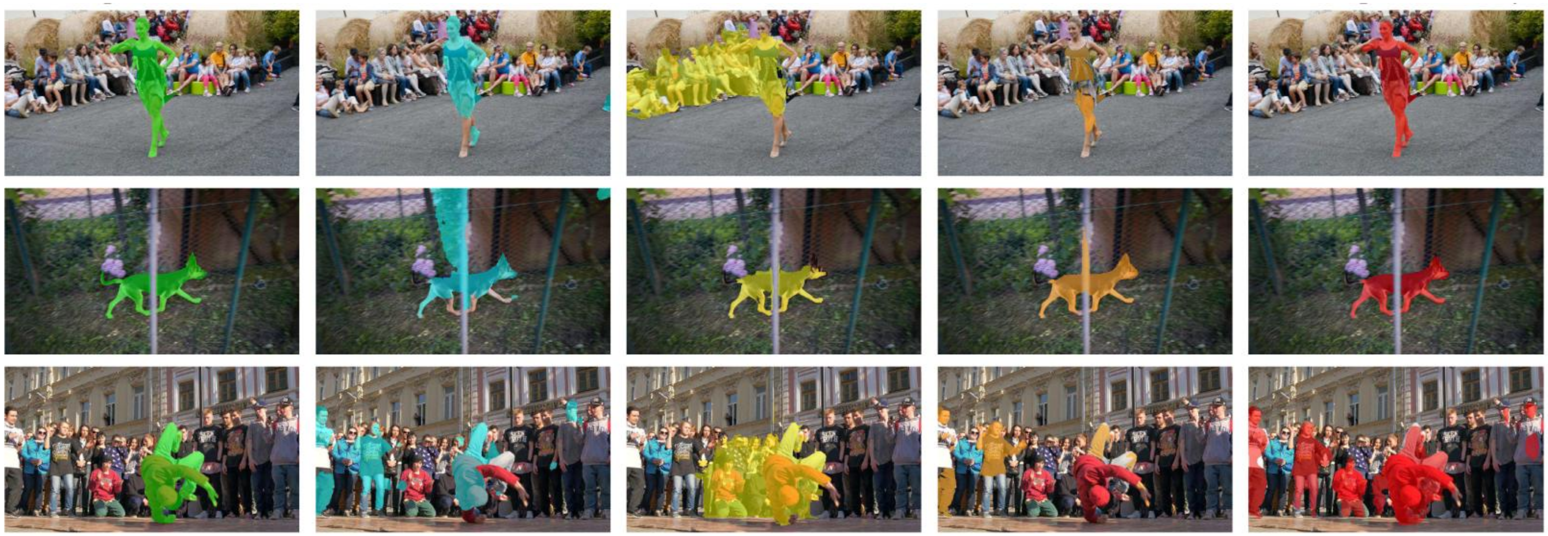
Figure 1. Sample results on the DAVIS dataset. Segmentations produced by MP-Net [43] (left) and our approach (right), overlaid on the video frame.

Comparison to MPNet

Method	Mean IoU
Ours	70.1
Ours + CRF	75.9
MP-Net	53.6
MP-Net + Obj	63.3
MP-Net + Obj + FST (MP-Net-V)	55.0
MP-Net + Obj + CRF (MP-Net-F)	70.0

Table 2. Comparison to MP-Net [43] variants on the DAVIS validation set. "Obj" refers to the objectness cues used in [43]. MP-Net-V(ideo) and MP-Net-F(rame) are variants of MP-Net which use FST [31] and CRF respectively, in addition to objectness.

Results - DAVIS



Ground truth | CUT [22] | FST [31] | MP-Net-Frame [43] | Ours

Figure 5. Qualitative comparison with top-performing methods on DAVIS. Left to right: ground truth, results of CUT [22], FST [31], MP-Net-Frame [43], and our method.

Results - DAVIS

Measure		PCM [3]	CVOS [41]	KEY [25]	MSG [4]	NLC [11]	CUT [22]	FST [31]	MP-Net-F [43]	Ours
$\mathcal{J}$	Mean	40.1	48.2	49.8	53.3	55.1	55.2	55.8	70.0	75.9
	Recall	34.3	54.0	59.1	61.6	55.8	57.5	64.9	85.0	89.1
	Decay	15.2	10.5	14.1	2.4	12.6	2.3	0.0	1.4	0.0
$\mathcal{F}$	Mean	39.6	44.7	42.7	50.8	52.3	55.2	51.1	65.9	72.1
	Recall	15.4	52.6	37.5	60.0	51.9	61.0	51.6	79.2	83.4
	Decay	12.7	11.7	10.6	5.1	11.4	3.4	2.9	2.5	1.3
$\mathcal{T}$	Mean	51.3	24.4	25.2	29.1	41.4	26.3	34.3	56.3	25.5

Table 3. Comparison to state-of-the-art methods on DAVIS with intersection over union ($\mathcal{J}$), F-measure ($\mathcal{F}$), and temporal stability ($\mathcal{T}$).

Results - DAVIS

Measure	ARP	LVO	FSEG	LMP	SFL	FST	CUT	NLC	MSG	KEY	CVOS	TRC
J Mean ↑	**76.2**	75.9	70.7	70.0	67.4	55.8	55.2	55.1	53.3	49.8	48.2	47.3
J Recall ↑	**91.1**	89.1	83.5	85.0	81.4	64.9	57.5	55.8	61.6	59.1	54.0	49.3
J Decay ↓	7.0	0.0	1.5	1.3	6.2	**0.0**	2.2	12.6	2.4	14.1	10.5	8.3
F Mean ↑	70.6	**72.1**	65.3	65.9	66.7	51.1	55.2	52.3	50.8	42.7	44.7	44.1
F Recall ↑	**83.5**	83.4	73.8	79.2	77.1	51.6	61.0	51.9	60.0	37.5	52.6	43.6
F Decay ↓	7.9	**1.3**	1.8	2.5	5.1	2.9	3.4	11.4	5.1	10.6	11.7	12.9
T (GT 8.8) ↓	39.3	26.5	32.8	57.2	28.2	36.6	27.7	42.5	30.1	26.9	**25.0**	39.1

Measure	OnAVOS	OSVOS	MSK	SFL	CTN	VPN	PLM	OFL	BVS	FCP	JMP	HVS	SEA
J Mean ↑	**86.1**	79.8	79.7	76.1	73.5	70.2	70.2	68.0	60.0	58.4	57.0	54.6	50.4
J Recall ↑	**96.1**	93.6	93.1	90.6	87.4	82.3	86.3	75.6	66.9	71.5	62.6	61.4	53.1
J Decay ↓	5.2	14.9	8.9	12.1	15.6	12.4	11.2	26.4	28.9	**-2.0**	39.4	23.6	36.4
F Mean ↑	**84.9**	80.6	75.4	76.0	69.3	65.5	62.5	63.4	58.8	49.2	53.1	52.9	48.0
F Recall ↑	89.7	**92.6**	87.1	85.5	79.6	69.0	73.2	70.4	67.9	49.5	54.2	61.0	46.3
F Decay ↓	5.8	15.0	9.0	10.4	12.9	14.4	14.7	27.2	21.3	**-1.1**	38.4	22.7	34.5
T (GT 8.8) ↓	19.0	37.8	21.8	18.9	22.0	32.4	31.8	22.2	34.7	30.6	15.9	36.0	**15.4**

Results - FBMS

Measure	Set	KEY [25]	MP-Net-F [43]	FST [31]	CVOS [41]	CUT [22]	MP-Net-V [43]	Ours
$\mathcal{P}$	Training	64.9	83.0	71.3	79.2	86.6	69.3	90.7
	Test	62.3	84.0	76.3	83.4	83.1	81.4	92.1
$\mathcal{R}$	Training	52.7	54.2	70.6	79.0	80.3	80.8	71.3
	Test	56.0	49.4	63.3	67.9	71.5	73.9	67.4
$\mathcal{F}$	Training	58.2	65.6	71.0	79.3	83.4	74.6	79.8
	Test	59.0	62.2	69.2	74.9	76.8	77.5	77.8

Table 4. Comparison to state-of-the-art methods on FBMS with precision ($\mathcal{P}$), recall ($\mathcal{R}$), and F-measure ($\mathcal{F}$).

CUT [22] MP-Net-Video [43] Ours

Figure 6. Qualitative comparison with top-performing methods on FBMS. Left to right: results of CUT [22], MP-Net-Video [31], and our method.

Results - SegTrack

CUT [22]	FST [31]	NLC [11]	Ours
47.8	54.3	67.2	57.3

Table 5. Comparison to state-of-the-art methods on SegTrack-v2 with mean IoU.