MED-VT: Multiscale Encoder-Decoder Video Transformer with Application to Object Segmentation
Supplemental Material

Rezaul Karim He Zhao Richard P. Wildes Mennatullah Siam
York University
{karimr31, zhufl, msiam}@eecs.yorku.ca, wildes@cse.yorku.ca

Abstract

This document provides additional material that is supplemental to our main paper. Section 1 describes the associated supplemental video. Section 2 provides further information on the technical approach as a description of the datasets, implementation details, and connection between the label propagation and spectral clustering. Section 3 provides additional empirical results in terms of ablations on the number of queries and the decoder learnable spatiotemporal embedding vs. *fixed sinusoidal spatiotemporal embedding as well as visualizations to shed further light on the contributions of the queries and label propagation. Section 4 details the used assets and accompanying licenses. Finally, Section 5 describes the societal impact of this research.*

1. Supplemental video

We include an accompanying supplemental video as part of the supplemental materials. In this video we show qualitative segmentation results of our approach. For AVOS we provide two examples on DAVIS'16 [10] and two examples on MoCA [4]. For actor/action segmentation we provide four examples on A2D [19]. The video is in MP4 format and approximately six minutes long. The codec used for the realization of the provided video is H.264 (x264). The video can be viewed at rkyuca.github.io/medvt

2. Technical details

This section unfolds in four subsections. First, we provide details for our pixel decoding scheme. Second, we establish the connection between label propagation and spectral clustering. Third, we describe the datasets used in our empirical evaluation. Fourth, we provide additional implementation and training details.

2.1. Pixel decoding details

In this subsection, we provide additional details on the operation of the Feature Pyramid Network (FPN) [5] used in our pixel decoding (main paper, Sec. 3.3). With our multiscale encoder having captured the recurring object and ensured temporally consistent features, the role of the FPN is to propagate high level semantics to the finest resolution feature maps. In the FPN, we combine the two coarsest resolution feature outputs from our multiscale encoder along with the two finest resolution outputs directly from the backbone. We only use the two coarsest resolutions in the multiscale encoder for memory efficiency reasons. The FPN operates on these four levels of resolution in a coarse-to-fine scheme by (i) performing a 1×1 convolutional operation that maps the features to channel dimension 384, (ii) performing ReLU and bilinear upsampling to match the next scale and (iii) pointwise adding the upsampled feature map to the next scale. The last three (coarsest) levels from the feature pyramid network output are used as input to our multiscale query learning mechanism, main paper, Sec. 3.3. Further details on how FPNs operate are available in the original paper [5].

2.2. Label propagation and spectral clustering

We begin with formal definitions from spectral clustering on which label propagation relies [15]. We include these here to establish the theoretical connection and help provide insights on how our label propagator operates. A similarity graph is defined in terms of a set of vertices, v_i and edges with weights, w_{ij}, on edges connecting vertices, v_i, and, v_j. The weight, w_{ij}, denotes the similarity between the two vertices; the similarity matrix, W, is comprised of the w_{ij}. For a vertex, v_i, we define the degree, $d_i = \sum_j^n w_{ij}$. The degree matrix, D, is defined as a diagonal matrix with degree $d_1, ..., d_n$, where each d_i sums the degree of connectivity for each vertex and its neighbours.

The normalized graph Laplacian matrix, which is a

corner stone of the label propagation [22], represents the smoothness of the graph function. It has two main forms, the symmetric and random walk, where the former was used in label propagation prior to deep learning [22]. Here, we work with the random walk graph Laplacian, as it aligns with transformer operations; it is defined as $\mathsf{L}_{rw} = I - \mathsf{D}^{-1}\mathsf{W}$. Since the identity matrix is constant, we can see the main component of the normalized graph Laplacian, $\mathsf{D}^{-1}\mathsf{W}$, corresponds to the calculated per head, h, attention map in our label propagation scheme, equation (8a) in the main paper, Softmax $\left(\frac{1}{\sqrt{\delta}}\mathsf{Q}\mathsf{W}_h^q(\mathsf{K}\mathsf{W}_h^k)^\top + \mathsf{M}\right)$. Here, we use δ for feature dimension to avoid confusion with d in defining the degree matrix, D.

The part of the attention operation that computes the relevance between the query and key tokens with masking that controls predefined neighbourhood, $\frac{1}{\sqrt{\delta}}\mathsf{Q}\mathsf{W}_h^q(\mathsf{K}\mathsf{W}_h^k)^\top$, corresponds to building the similarity matrix, W. We compute the similarity between two nodes in the similarity matrix, W, in terms of a scaled inner product, as standard in multihead attention. The masking operation, M, serves to further enforce cliques within the similarity graph based on the structure of the data (*e.g*. the temporal structure). On the other hand, the normalization of the graph Laplacian using the degree matrix, D^{-1}, corresponds to the Softmax normalization in the attention operation. Overall, given that the attention map in equation (8a) when applied to our values, V, controls the label propagation, we see that it acts analogously to the way the graph Laplacian enforces smoothness.

2.3. Datasets

In this section, we provide additional details on the three standard Automatic Video Object Segmentation (AVOS) dataset used to evaluate our model. These are DAVIS'16 [10], YouTube-Objects [11] and MoCA (Moving Camouflaged Animals) [4]. We also provide additional details on the actor/action segmentation dataset that was used in our evaluation, A2D [19].

DAVIS'16 consists of 50 video sequences with a total 3455 annotated frames of 480p and 720p with high quality dense pixel-level annotations (30 videos for training and 20 for testing). For quantitative evaluation on this dataset, we follow the standard evaluation protocol provided with the dataset [10]. We report results as mean Intersection over Union (mIoU), $\mathcal{J}$, and boundary accuracy, $\mathcal{F}$, using 480p resolution frames.

YouTube-Objects contains 126 videos with more than 20,000 frames at resolution 720p. Following its protocol, we use mIoU to measure performance. We follow standard protocol by training on DAVIS'16 and evaluating on all videos in YouTube-Objects dataset and evaluate per category intersection over union and report the average.

MoCA consists of videos of moving camouflaged animals with more than 140 clips of resolution 720p across a diverse range of animals. This is the most challenging motion segmentation dataset currently available, as in the absence of motion the camouflaged animals are almost indistinguishable from the background by appearance alone (*i.e*. colour and texture). The groundtruth is provided as bounding boxes, and we follow previous work [21] in removing videos that contain no predominant target locomotion, to evaluate on a subset of 88 videos constituting approximately 2803 frames. We evaluate using mIoU with the largest bounding box on the predicted segmentation and also report the success rate with varying IoU thresholds, $\tau \in \{0.5, ..., 0.9\}$. Following standard practice [23], we trained our model on DAVIS'16 and YouTube-VOS.

A2D consists of 3782 videos from YouTube that were annotated for nine different actors and seven actions, the final valid categories of actor-action pairs constitute 43 pairs. The dataset has video frames with resolution 320p and it contains training and testing subsets. Following its protocol, we evaluate mean intersection over union on the actor-action categories in the test set.

2.4. Implementation, training and inference details

In this section, we provide additional implementation and training details not described in the main paper.

Implementation details. We extract backbone features, F, with feature activations from the output of each stage's last block, which have strides resp. 4, 8, 16, and 32 for ResNet-101 and 8, 16, 32, 32 for Video-Swin. The features are used as our initial multiscale features, f_s. The task head, $\mathcal{H}$, in both the AVOS and actor/action segmentation tasks is instantiated as a small 3D convolutional network consisting of three layers of convolution with group normalization [18] and the ReLU activation function. We use clip length, $T = 6$ during training and inference; correspondingly, we set the number of learnable queries to, $N_q = 6$.

Training and inference details. During training, the input images are downsampled so that the smallest side length becomes 300. We employ data augmentation strategies, including vertical and horizontal flipping as well as multiscale training [12]. Following MATNet [23], we use the training data in DAVIS'16 [10] and the training set of YouTube-VOS [20] resulting in 14K images. We use AdamW [7] as optimizer, with initial learning rate 10^{-6} for the backbone and 10^{-4} for the rest of the model. We use weight decay of 1×10^{-4} and train for 210,000 iterations using polynomial learning rate decay with power 0.9. We initialize the model from COCO pretrained weights [6], following [1], and we freeze the batch normalization layer weights for the backbone network. We follow a two stage training strategy for AVOS: (i) We train our model without label propagation; (ii) we freeze the weights and train the label propagator. During inference on DAVIS'16 only, we use multiscale inference for postprocessing with predictions averaged over

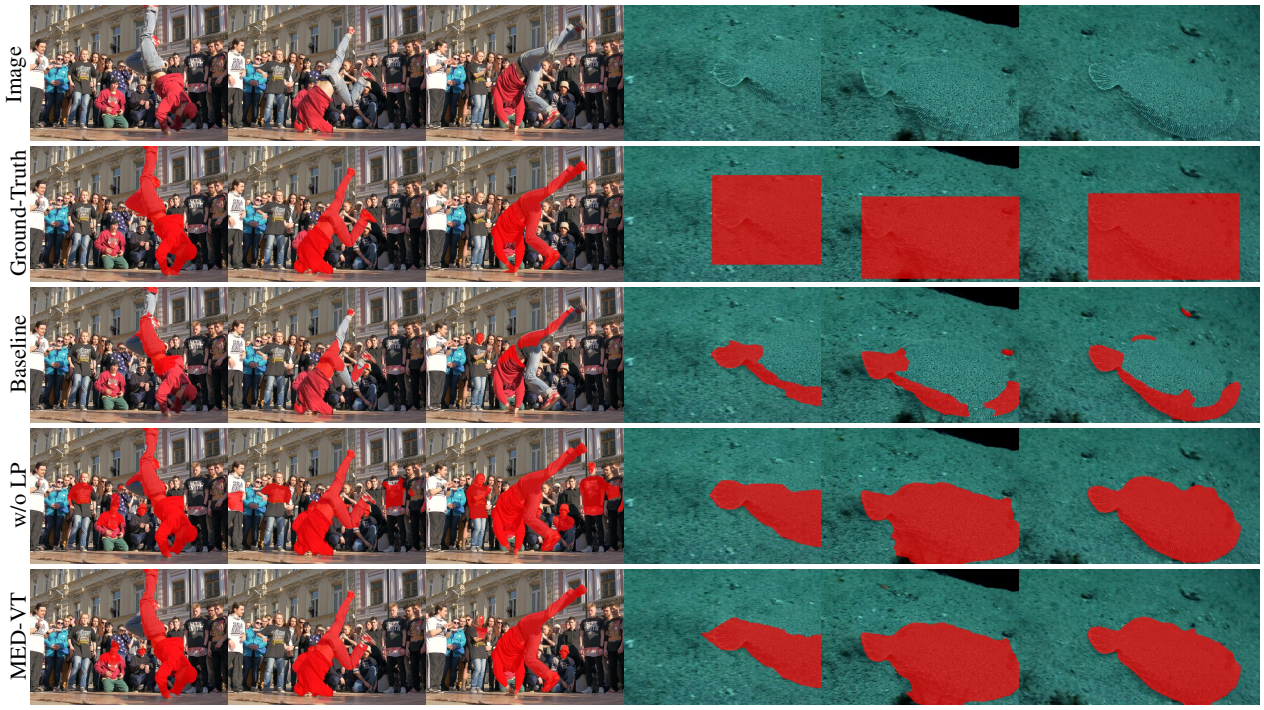

Figure 1. Qualitative segmentation results (red masks) showing the efficacy of our full model and specifically the label propagation module. From top to bottom, rows are arranged as input image, ground truth, our single scale encoder-decoder (*baseline*), MED-VT w/o label propagation (*w/o LP*), and MED-VT. **Left:** Three frames of DAVIS'16 breakdance. **Right:** Three frames of MoCA flounder-6. MED-VT w/o label propagation improves over single scale baseline in identifying the prominent object. Label propagation reduces false positives (breakdance) and also improves object boundaries and reduces small false negatives (flounder-6).

different scales using scale multipliers, (0.7, 0.8, . . . , 1.2). As discussed in the main paper, it is standard practice to show results on DAVIS with and without postprocessing.

For actor/action segmentation, we follow the same settings as for AVOS, except that we train for 46,000 iterations and we found it necessary to adopt a three stage learning strategy: (i) We train the single scale encoder MED-VT without label propagation; (ii) we train the multiscale encoder with between scale attention and freeze the rest; (iii) we train the label propagation module. Recall that the A2D dataset is harder than the AVOS datasets, because it encompasses considerably more categories (*e.g.*, 43 action-actor tuples) yet far fewer annotations (*e.g.*, three frame-level annotations per video); therefore, end-to-end training for all modules has proven to be difficult. All our models on both tasks are implemented using the PyTorch library [9] and were run on an NVIDIA Quadro P6000 GPU.

3. Additional empirical results

In this section we provide additional empirical results. First, we show qualitative results to demonstrate our model's temporal consistency on the prediction level via comparing with and without label propagation, and on the feature level via visualizing the first three principal components of the features. We also provide a quantitative analysis of temporal consistency by plotting IoU vs. time for two example videos. Second, we provide per category results for YouTube-Objects. Third, we study via visualizations the nature of object attention maps output from the multiscale query learning and attention block, $\mathcal{A}^D$. Fourth, we investigate using a single query for the entire input clip *vs*. a single query per frame in the input clip. Fifth, we compare using learnable query positional embedding *vs*. static sinusoidal query positional embeddings per scale, p_s^Q.

3.1. Temporal consistency results

Prediction level temporal consistency. We present a qualitative ablation of temporal consistency on the prediction level using our label propagation with a ResNet101 backbone in Fig 1. It is seen in the left three columns that the single scale encoder decoder (*baseline*) fails to segment objects under deformable motion surrounded by background objects of the same semantic class (*i.e.* human). Although multiscale encoder-decoder without label propagation (*w/o LP*) is seen to identify the prominent object, there are some false positives due to the presence of background objects belonging to the same semantic class. However, the

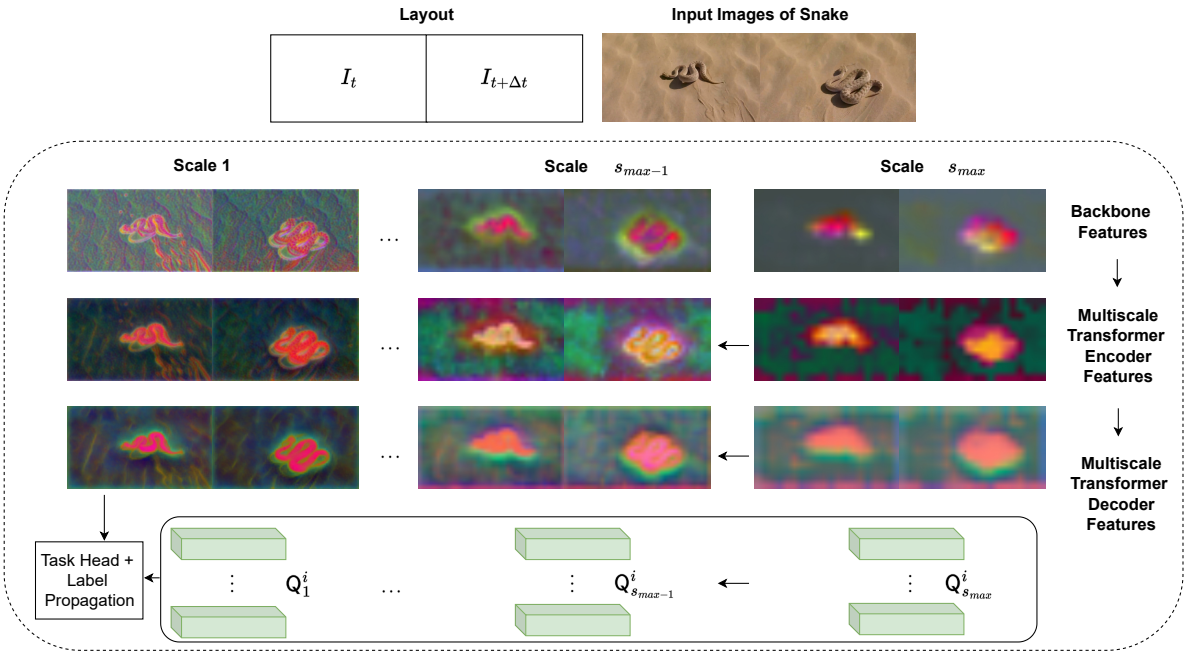

Figure 2. Our unified Multiscale Encoder-Decoder Video Transformer (MED-VT) with visualization of the intermediate representations learned. For the sake of visualization, we use PCA on the intermediate feature maps and show the first three components indicated by RGB. Input is a clip of N frames where we show only two RGB frames for visualization, *i.e.* no optical flow input. Backbone features show the fine scale feature map has precise details, but suffers from noise as it captures only low level semantics; in comparison, coarser scales have better abstraction (*e.g.* overall shape of the snake is highlighted), but suffer from lack of precise localization. Multiscale encoding improves spatiotemporal consistency and focuses on the recurring object. In complement, multiscale decoding provides better foreground localization via learning multiscale adaptive queries, Q^i_s, for iteration, i, and scale, s. The fine scale queries and decoded feature maps are input to our task specific head and label propagation. Arrows show direction of information flow during processing.

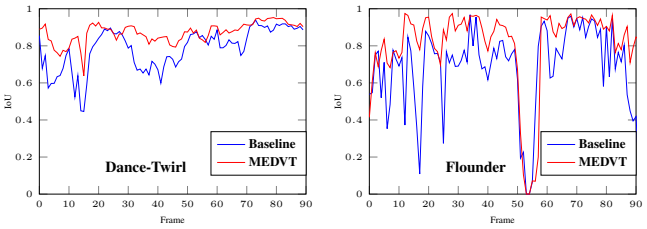

Figure 3. Temporal consistency analysis using IoU over time in dance-twirl and flounder videos in Fig.4 main paper. Flounder IoU drop around frame 53 due to sudden large camera motion.

label propagation in our full model (*MED-VT*) reduces the false positive segmentation substantially without damaging the foreground object by means of many-to-many temporal propagation of initial labels. The right three columns show three frames of MoCA flounder-6. This example demonstrates how our many-to-many label propagation improves object boundaries and reduces false negatives, as the final segmentation mask more precisely delineates the shape of the fish.

As another demonstration of temporal consistency, we provide IoU over time results in Figure 3. It is clear that our results are comparably smoother than the baseline, *e.g.* show less zig-zag pattern, and thereby document better temporal consistency in the segmentation.

Feature level temporal consistency. We now present qualitative results to demonstrate the temporal consistency of our model on the feature level. Here, we focus on the components within our integrated multiscale encoder-decoder without label propogation, as the benefits of label propagation were demonstrated in the previous experiment. For visualization, we compute the first three principal components, indicated by RGB, for the features at different scales. Figure 2 shows that our multiscale encoding with between scale attention, $\mathcal{B}$, improves spatiotemporal consistency and focuses on the recurring object. In complement it also is seen that our multiscale decoding supports capture of the foreground object, by projecting the recurring temporally consistent object learned at the coarser resolutions to the finer resolutions. These operations enable the delineation of fine-grained object details before going through the task specific head and temporal label propagation.

3.2. YouTube-Objects per category results

Table 1 shows per category results on the YouTube-Objects dataset. It is seen that our approach outperforms

previous state-of-the-art AVOS approaches, even with the weaker ResNet-101 backbone and without using extra optical flow input. This observation is especially evident in the most challenging category, "Train", where our approach outperforms the previous state-of-the-art approach that uses optical flow input by up to 16%.

3.3. Contribution of adaptive queries

To analyze the object attention maps generated by the learnable queries of the MED-VT decoder, we use visualization. Figure 4 shows the object attention generated by the attention block, $\mathcal{A}^D$, per attention head. We observe that different attention heads learn to attend to different aspects of object representation, including the object extremities, and that most of the attention heads generate well localised saliency maps for the primary objects in the video. It is interesting to observe that while some of the attention heads generate very good localization of the overall primary object (*e.g.* H_4, H_5, H_8), others generate mainly object boundaries (*e.g.* H_2). Additionally, some of the attention heads instead generate complementary saliency for the background (*e.g.* H_7). These results provide insight on the nature of attention generated through use of decoder queries, which has not been presented in such detail in previous work on transformer decoders. Furthermore, we analyze the discriminating attention produced by all the heads combined, *i.e.* the object attention map $F^{\mathcal{A}}$; see Fig 5. For this purpose, we compute the first three principal components of our output attention maps on all heads and plot them in a single image with RGB format. These results confirm the localization ability of our adaptive foreground queries that improve with the multiscale query learning in a coarse-to-fine processing manner.

3.4. Per-clip *vs*. per-frame queries

The quantitative results of our additional ablation experiment on the number of queries, N_q, are shown in Table 2. As expected, the per-frame foreground queries surpass the per-clip queries by 1.2% on DAVIS'16 and MoCA. The object motion and deformation that occurs during a video entails that learning per-frame queries will generate better attention maps than per-clip. This ability can lead to highlighting different positions and extremities of the primary object parts in the input clip. It is important especially in MoCA, which suffers from the object boundaries deforming and blending with the background.

3.5. Query positional embedding

Quantitative results for our ablation experiment on static vs. learnable position embedding for the dynamic object queries are shown in Table 3. It is seen that learnable positional embeddings for the query perform better than sinusoidal positional embeddings, with an especially notable margin on a benchmark that focus on motion rather than object appearance (*i.e.* MoCA).

3.6. Training and Inference Efficiency

We perform all benchmarking experiments on a Linux Server equipped with Intel(R) Xeon(R) W-2155 3.30GHz CPU and an NVIDIA Quadro P6000 24 GB GPU. We run all experiments using a single GPU. The training time for our model is approximately two days. We run other models using their publicly available code and trained models.

Table 4 shows a comparison of inference time, memory use and mIoU score on the MoCA dataset. It is seen that our approach is indeed efficient as it achieves the best performance while being faster in run time. This efficiency is a result of our approach not depending on time consuming optical flow estimation and the parallelization enabled by our ability to process a video clip as a whole rather than sequentially frame-by-frame.

Due to the use of a multiscale transformer encoder, our approach requires more GPU memory than other approaches. An interesting future direction is to investigate memory efficient video transformers, *e.g.* [3].

4. Licenses and assets

We use the DAVIS'16 [1] and YouTube-VOS [2] datasets during training, where both are licensed under a Creative Commons Attribution 4.0 License that allows non-commercial research use. Additionally, we use YouTube-Objects [3] and MoCA [4] for evaluation, which are under the same license. We also used A2D[5] dataset with a license that prevents republishing of the dataset without authors consent.

5. Societal impact

Automatic video object segmentation and actor/action segmentation have multiple positive societal impacts as they can be used for a variety of useful applications, *e.g.* autonomous driving and robot navigation. The class agnostic segmentation of moving objects and actors in autonomous driving can encourage safety critical decision making and reduce accidents from unknown objects outside the closed set of classes predefined in large-scale datasets. Their use in robot navigation can also serve a wide variety of impactful applications such as human-robot interactions, improving health care systems and elder care.

However, as with many AI abilities, automatic video object and actor/action segmentation also can have negative

[1] https://davischallenge.org/davis2016/code.html
[2] https://youtube-vos.org/dataset/
[3] https://data.vision.ee.ethz.ch/cvl/youtube-objects/
[4] https://www.robots.ox.ac.uk/~vgg/data/MoCA/
[5] https://web.eecs.umich.edu/~jjcorso/r/a2d/

Category	Uses RGB+Flow				Uses RGB only					
	FSEG [2]	LVO [14]	MATNet [23]	RTNet [12]	PDB [13]	AGS [17]	COSNet [8]	AGNN [16]	**Ours**	**Ours†**
Airplane(6)	81.7	86.2	72.9	84.1	78.0	87.7	81.1	81.1	**88.9**	88.3
Bird(6)	63.8	**81.0**	77.5	80.2	80.0	76.7	75.7	75.9	73.0	80.7
Boat(15)	72.3	68.5	66.9	70.1	58.9	72.2	71.3	70.7	77.2	**79.4**
Car(7)	74.9	69.3	79.0	79.5	76.5	78.6	77.6	78.1	77.3	**86.8**
Cat(16)	68.4	58.8	73.7	71.8	63.0	69.2	66.5	67.9	79.6	**82.2**
Cow(20)	68.0	68.5	67.4	70.1	64.1	64.6	69.8	69.7	**76.2**	75.8
Dog(27)	69.4	61.7	75.9	71.3	70.1	73.3	76.8	77.4	75.7	**80.8**
Horse(14)	60.4	53.9	63.2	65.1	67.6	64.4	67.4	67.3	**68.7**	67.2
Motorbike(10)	62.7	60.8	62.6	64.6	58.3	62.1	67.7	68.3	65.9	**69.2**
Train(5)	62.2	66.3	51.0	53.3	35.2	48.2	46.8	47.8	69.3	**73.5**
Mean	68.4	67.5	69.0	71.0	65.4	69.7	70.5	70.8	75.2	**78.5**

Table 1. Results of object class segmentation on YouTube-Objects. Results shown as mean Intersection over Union (mIoU) per category as well as overall average across all categories. † indicates our model with Video-Swin backbone. Best results higlighted in **bold**.

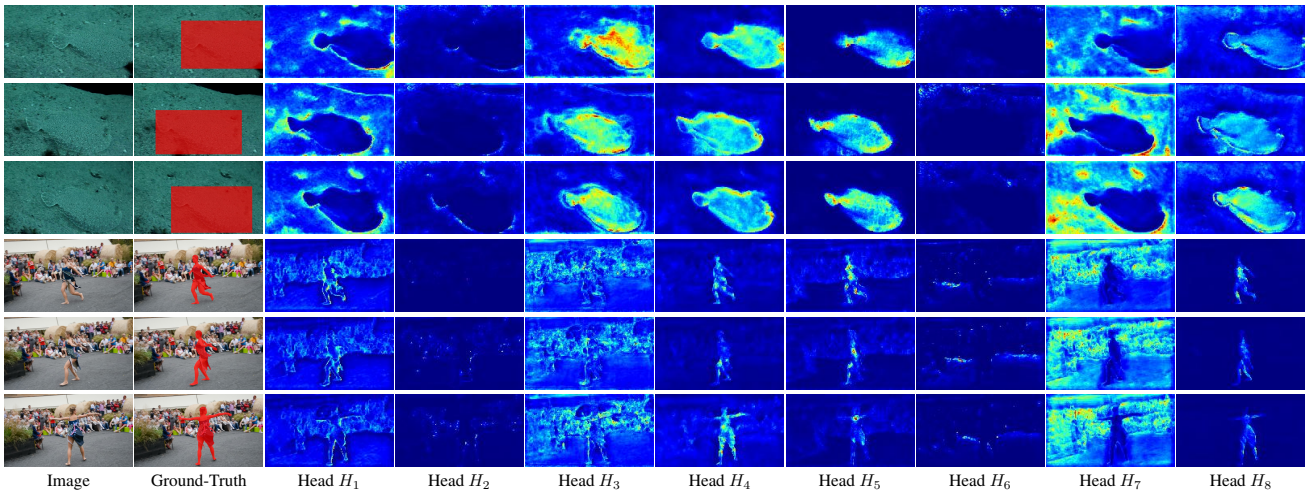

Figure 4. Object attention generated by $N_h = 8$ different attention heads based on the dynamic query for randomly selected frames from two video sequences. **Top:** three rows show attention maps for the flounder-6 video from MoCA dataset. **Bottom:** three rows show attention maps for dance twirl video from DAVIS'16. Attention maps are shown as a heat-map using jet color-space.

Number of query	DAVIS'16	MoCA
Per clip ($N_q = 1$)	81.8	68.2
Per frame ($N_q = T$)	**83.0**	**69.4**

Table 2. Ablation on the number of queries reporting mIoU. T is equal to the number of frames in an input clip. Best results highlighted in **bold**.

Decoder Query Position Embedding	DAVIS'16	MoCA
Static Sinusoidal	80.06	65.4
Learnable	**83.0**	**69.4**

Table 3. Learnable position embedding vs static sinusoidal position embedding for query reporting mIoU. Best results highlighted in **bold**.

societal impacts, *e.g.* through application to automatic target detection in military systems and falsification of video

Model	Time	mIoU	Memory
MATNet [54]	0.9	64.2	**2577**
RTNet [32]	1.9	60.7	3615
COSNet [23]	1.3	50.7	9255
MEDVT (ours)	**0.6**	**69.4**	9509

Table 4. Comparison on run time and memory consumption. Best results highlighted in textbfbold.

documents via object removal. To some extent, movements are emerging to limit such applications, *e.g.* pledges on the part of researchers to ban use of artificial intelligence in weaponry systems. We have participated in signing that pledge and are supporters of its enforcement through international laws.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the*

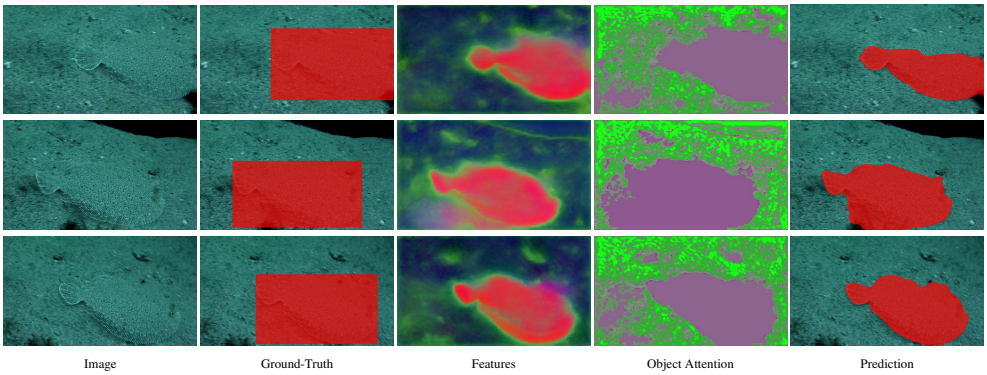

Figure 5. Visual summary of object localization using our adaptive foreground queries in multiscale query learning. The object attention maps, $F^{\mathcal{A}}$, result from combining all attention heads; visualized as the first three principal components of each attention map plotted as an RGB image. Top to bottom: Three frames of flounder-6 from MoCA dataset. Left to right: Input images, bounding box masks, decoder features, $f_1^{\mathcal{P}}$, object attention, $F^{\mathcal{A}}$, and the final prediction.

European Conference on Computer Vision, pages 213–229, 2020. 2

[2] Suyog Dutt Jain, Bo Xiong, and Kristen Grauman. FusionSeg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2117–2126, 2017. 6

[3] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations*, 2020. 5

[4] Hala Lamdouar, Charig Yang, Weidi Xie, and Andrew Zisserman. Betrayed by motion: Camouflaged object discovery via motion segmentation. In *Proceedings of the Asian Conference on Computer Vision*, pages 488–503, 2020. 1, 2

[5] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017. 1

[6] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, pages 740–755, 2014. 2

[7] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 2

[8] Xiankai Lu, Wenguan Wang, Chao Ma, Jianbing Shen, Ling Shao, and Fatih Porikli. See more, know more: Unsupervised video object segmentation with co-attention siamese networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3623–3632, 2019. 6

[9] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*, volume 32, 2019. 3

[10] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 724–732, 2016. 1, 2

[11] Alessandro Prest, Christian Leistner, Javier Civera, Cordelia Schmid, and Vittorio Ferrari. Learning object class detectors from weakly annotated video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3282–3289, 2012. 2

[12] Sucheng Ren, Wenxi Liu, Yongtuo Liu, Haoxin Chen, Guoqiang Han, and Shengfeng He. Reciprocal transformations for unsupervised video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15455–15464, 2021. 2, 6

[13] Hongmei Song, Wenguan Wang, Sanyuan Zhao, Jianbing Shen, and Kin-Man Lam. Pyramid dilated deeper convlstm for video salient object detection. In *Proceedings of the European Conference on Computer Vision*, pages 715–731, 2018. 6

[14] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning video object segmentation with visual memory. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4481–4490, 2017. 6

[15] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007. 1

[16] Wenguan Wang, Xiankai Lu, Jianbing Shen, David J Crandall, and Ling Shao. Zero-shot video object segmentation via attentive graph neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9236–9245, 2019. 6

[17] Wenguan Wang, Hongmei Song, Shuyang Zhao, Jianbing Shen, Sanyuan Zhao, Steven CH Hoi, and Haibin Ling. Learning unsupervised video object segmentation through

visual attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3064–3074, 2019. 6

[18] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision*, pages 3–19, 2018. 2

[19] Chenliang Xu, Shao-Hang Hsieh, Caiming Xiong, and Jason J Corso. Can humans fly? action understanding with multiple classes of actors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2264–2273, 2015. 1, 2

[20] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. YouTube-VOS: Sequence-to-sequence video object segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 585–601, 2018. 2

[21] Charig Yang, Hala Lamdouar, Erika Lu, Andrew Zisserman, and Weidi Xie. Self-supervised video object segmentation by motion grouping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7177–7188, 2021. 2

[22] Dengyong Zhou, Olivier Bousquet, Thomas Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*, volume 16, 2003. 2

[23] Tianfei Zhou, Shunzhou Wang, Yi Zhou, Yazhou Yao, Jianwu Li, and Ling Shao. Motion-attentive transition for zero-shot video object segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13066–13073, 2020. 2, 6