EVA-VOS: Efficient Video Annotation for Video Object Segmentation

Thanos Delatolas[1] Vicky Kalogeiton[2] Dim P. Papadopoulos[1]
[1] Technical University of Denmark [2] LIX, Ecole Polytechnique, CNRS, Institut Polytechnique de Paris
atde@dtu.dk, vicky.kalogeiton@lix.polytechnique.fr, dimp@dtu.dk

Abstract

Training a Video Object Segmentation (VOS) model requires an abundance of manually labelled training videos. The de-facto way of annotating objects requires humans to draw detailed segmentation masks on the target objects at each frame, which is tedious and time-consuming. To reduce this annotation cost, we propose EVA-VOS, a human-in-the-loop Efficient Video Annotation framework for VOS. Unlike de-facto approaches, we introduce an agent that predicts iteratively both which frame to annotate and which annotation type to use. Then, the annotator annotates only the selected frame that is used to update a VOS module, leading to significant gains in annotation time. We experiment on the MOSE dataset and show that: (a) EVA-VOS leads to masks with accuracy close to the human agreement 3.5× faster than the standard way of annotating videos; (b) our frame selection achieves state-of-the-art performance; (c) EVA-VOS yields significant performance gains in terms of annotation time compared to all methods and baselines. Code data and models are available online[1]

1. Introduction

Video object segmentation (VOS) is the task of segmenting and tracking objects in videos [6, 48, 22, 42, 21, 43, 29, 31, 13, 12, 49, 50, 11, 4]. VOS is a central task for video understanding and enables various applications including video editing [23, 2], synthesis [44, 45], and decomposition [51]. Training a VOS model requires videos where the target objects have been manually annotated with object segmentation masks [6, 31, 29, 13, 12, 11, 4, 49, 50]. This process is expensive as it requires humans to manually draw a mask at each video frame, requiring 80 seconds per object per frame [28]. For instance, annotating only one object in a 10-second video would require more than 5 hours.

To address these, two groups of solutions are adopted. The first solution consists in sparsely annotating large VOS datasets [47, 38, 16, 15, 46]. The standard way of annotating a VOS dataset [36, 37, 47, 46, 15, 41, 38, 16] starts by uniformly sampling a subset of frames (1-5 fps) and then each frame is manually annotated with a mask by an annotator. In some datasets [15, 46], the sparsely annotated masks are interpolated to predict dense annotations.

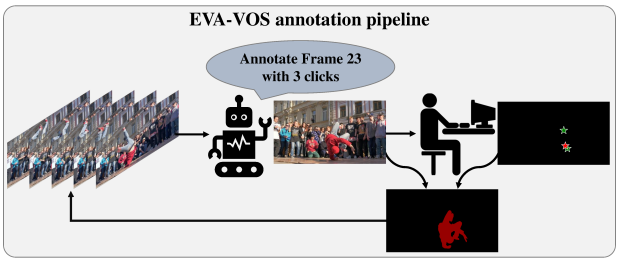

Figure 1: **EVA-VOS.** In contrast to the traditional way of annotating objects in videos, we propose to use a human-in-the-loop approach. We introduce an agent that selects the frame that should be annotated and the annotation type (e.g. clicks, object-mask). Then, we use the weak annotation to predict a mask for the frame and we propagate it to predict masks for the whole video.

The second solution consists in minimizing the annotation cost, typically by interactive segmentation using faster annotation types, such as clicks, scribbles, or bounding boxes. Albeit the progress on still images [1, 5, 8, 27, 30, 33, 35] with large datasets [5], there is limited work for videos [7, 9, 10, 14]. The most relevant to our work is [7] that propose a human-in-the-loop interactive VOS. The annotator provides a scribble and a VOS method predicts a mask for each frame. Then, the annotator iteratively selects the frame with the worst segmentation quality and provides scribbles. Despite its novelty, it has two limitations. First, it is unrealistic as the annotator cannot identify the worst frame; even if they could, it would require significant time [20], which defeats the cost minimization. Second, in challenging frames, low-cost annotation types cannot create good masks, and drawing the full mask is required.

To overcome these, we propose **EVA-VOS**, an **E**fficient **V**ideo **A**nnotation pipeline for **V**ideo **O**bject **S**egmentation (Fig. 1). We introduce an agent that predicts iteratively which *frame* should be annotated (frame selection) and which *annotation type* to use (annotation selection). Our agent is trained to maximize the annotation impact on the

[1] https://eva-vos.compute.dtu.dk/

segmentation quality while minimizing the annotation cost. For frame selection, we train a model to regress the quality of a segmentation mask. Then, we select the frame with the maximum distance from its closest pre-annotated frame. For annotation selection, we train a deep RL policy to select an annotation type (action) by maximizing the fraction of the segmentation quality improvement over the annotation time of the annotation type (reward). Our method iterates between (a) selecting the next frame for annotation and the annotation type, (b) asking annotators to improve a segmentation mask, and (c) predicting new object masks for all frames (Fig. 2). We experiment on the MOSE [16] dataset. We evaluate both each stage independently and our full pipeline. We show that (a) EVA-VOS leads to masks with accuracy close to human agreement 3.5× faster than the standard way of annotating a VOS dataset; (b) Our frame selection method achieves state-of-the-art performance; (c) EVA-VOS yields significant performance gains in terms of annotation time compared to other strong baselines.

2. Method

We propose EVA-VOS, a human-in-the-loop pipeline to annotate videos with segmentation masks using as little annotation as possible (Fig. 2). EVA-VOS consists of four stages: (a) Mask Propagation (Sec. 2.1), (b) Frame Selection (Sec. 2.2), (c) Annotation Selection (Sec. 2.3), (d) Annotation and Mask Prediction (Sec. 2.4).

More formally, at each iteration t, the mask propagation receives the input video $V = \{f_1, f_2, \dots, f_N\}$ of N frames and a set K containing all previously annotated frames to predict a new set of masks $\mathbf{M}^t = \{M_1^t, M_2^t, \dots, M_N^t\}$ for all frames. Then, the frame selection determines the frame f_* that should be annotated given V and $\mathbf{M}^t$. The annotation selection determines the most suitable annotation type a_{f_*} from a pool of candidate annotation types $A = \{a_1, a_2, \dots, a_L\}$. For annotation types, we consider both the case where the annotator manually draws a complete mask (*'mask drawing'*), and the case of weak annotations, where the human intervention is much faster, e.g., *'corrective clicks'*, *'bounding boxes'*, *'scribbles'*, etc. Finally, the annotator annotates f_* with a_{f_*}, and the annotation is passed on to the mask prediction, where a new mask $M_{f_*}^{t+1}$ is predicted and added to K. Note that at $t = 0$, the annotator selects the target object and draws a mask on f_1.

2.1. Mask propagation

We predict masks $\mathbf{M}^t$ for all frames using all annotated masks from K. We use a pre-trained VOS [12] with inputs video V and masks K and output mask M_i for each frame.

2.2. Frame selection

Given V, $\mathbf{M}^t$, and K, we aim at finding the frame to be annotated f_* at iteration t with the highest improvement on the video segmentation quality at iteration $t+1$. Intuitively, we select frames that maximize the diversity among the selected ones while having low segmentation quality. We train a model to assess the segmentation quality of each frame, and then use the learned frame representations to select f_*.

Architecture. To access the mask quality of each frame, we propose the Quality Network (QNet) which takes in a frame f_i and its mask M_i^t and performs mask quality classification into B classes, where 0 represents the worst quality and $B-1$ the best. B determines the number of bins of the segmentation quality. QNet consists of two image encoders [19] in parallel branches, one for frame f_i and one for mask M_i^t. The embeddings from each encoder are concatenated and fed into a linear classifier of B outputs.

Training. We train QNet in a supervised way with cross-entropy loss on a simulated training set. To generate it, we simulate a number of iterations with EVA-VOS (Fig. 2); at each iteration, we compute the segmentation quality of each frame and assign a quality label to each mask M_i^t. To further augment it, we include random selections.

Selected frame. f_* is the one with the maximum distance in the feature space from its closest previously annotated frame. We (1) extract embeddings E_i from QNet; (2) compute the L2 between each embedding of frame j in K and frames of V (3) assign the minimum distance to each embedding i, and select the frame with the maximum distance:

$$f_* = \underset{i \in \{1,2 \dots N\}}{\arg\max} \left\{ \min_{j \in \{1,2 \dots t\}} \{d(E_i, E_j)\} \right\} \quad (1)$$

2.3. Annotation selection

Given annotation types $A = \{a_1, a_2, \dots, a_L\}$, this step chooses the most suitable type a_{f_*} for f_*. We formulate this as a Markov Decision Process and train a model using reinforcement learning (RL). The model observes the image of f_* and its predicted mask $M_{f_*}^t$ and predicts the most suitable annotation type a_{f_*}. This a_{f_*} is used by the annotator to generate a new mask $M_{f_*}^{t+1}$ for f_*. The annotation is performed iteratively (e.g. 3 clicks are performed one by one). Therefore, we denote the annotation iteration as g. The input $M_{f_*}^t$ has an initial segmentation quality SQ_1 ($g = 1$).

Environment. The state of the environment consists of f_* and its mask. Each step g yields SQ_g using the input action, which represents an annotation type from A.

Reward. The reward function reflects the trade-off between the quality of $M_{f_*}^t$ and the cost of the annotation type. Each $a \in A$ requires a different annotation cost θ_a. The reward at g is formulated by comparing SQ before and after annotation, divided by the total cost tc at g which is the sum of the costs θ_a of all annotation types until g: $r = \frac{\text{SQ}_{g+1} - \text{SQ}_g}{tc}$.

Architecture. The model has two image encoders [26, 19] in parallel branches, one for the frame f_i and one for the mask $M_{f_*}^t$. The extracted embeddings from each encoder are then concatenated and fed into two linear layers. The

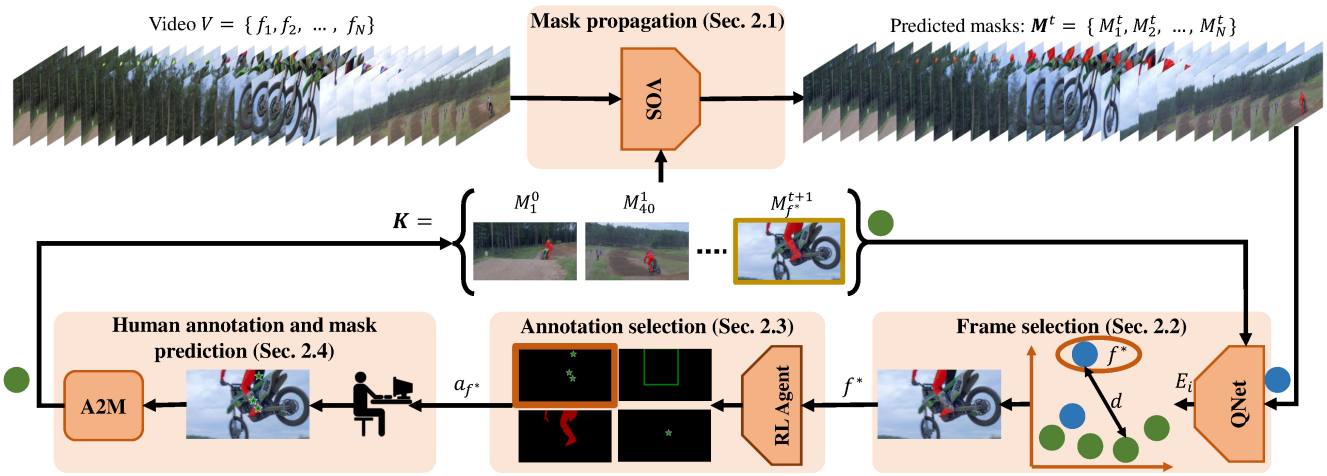

Figure 2: **EVA-VOS**. At each iteration t, Mask propagation (Sec. 2.1) receives a video V of N frames and a set K containing all previously annotated frames to predict a new set of masks $\mathbf{M}^t = \{M_1^t, M_2^t, \ldots, M_N^t\}$ for all frames. Subsequently, the Frame selection (Sec. 2.2) stage selects the frame f_* that should be annotated given the video V, the predicted masks $\mathbf{M}^t$ and all previously annotated frames K. The Annotation selection (Sec. 2.3) takes as input the selected frame f_* and its corresponding mask to predict the most suitable annotation type a_{f_*}. Finally, in the Human annotation and mask prediction (Sec. 2.4) stage, the annotator interacts with f_* using the annotation type a_{f_*} and A2M (in this work, we use SAM [26]) predicts the new mask $M_{f_*}^{t+1}$ of the frame f_*, which is then added to the set K.

first layer has L outputs (possible annotation types), while the second layer has one output for the RL value.

Training. We use Proximal Policy Optimization [40] (PPO) to train our model. At training, we use the simulated masks described in Sec. 2.2. We perform multiple environment steps and the process terminates when we reach the maximum steps or the type of drawing a mask is selected.

Video Ranking. We use the predicted value of our RL agent to estimate the improvement of each annotation at each video. This enables ranking the videos and performing more annotation iterations in videos with higher RL value.

2.4. Human annotation and mask prediction

The annotator interacts with the f_* to create the input a_{f_*}. When a_{f_*} is *'mask drawing'*, the annotator draws a detailed $M_{f_*}^{t+1}$. Otherwise, i.e. clicks, this step predicts a new mask $M_{f_*}^{t+1}$ using a pre-trained A2M model. Since EVA-VOS is independent of this model, we opt for the recently introduced Segment Anything Model (SAM) [26].

3. Experimental setting

Datasets. We use the MOSE dataset [16] which contains 1507 videos with available ground-truth segmentation masks. We only consider videos with 15 to 104 frames leading to a MOSE-long with 1166 videos. We split it into 800 training, 150 validation, and 216 test videos.

Metrics. We use the curve of $\mathcal{J}\&\mathcal{F}$ vs time [7] and the annotation time at $\mathcal{J}\&\mathcal{F} = 0.85$ (human annotation agreement for instance segmentation [5, 18, 25, 53]). We consider 80 sec for drawing a mask [28] and 1.5 sec for each click plus 1 sec of overhead to locate the object [3, 5, 34].

Implementation details. QNet consists of two ResNet-18 [19]. We train it using SGD with lr=10^{-5}, batch size 64, 30 epochs, with B=20. The frame branch of the RL agent is the image encoder of SAM [26]; the mask branch is ResNet-18 [19]. The RL agent is trained using Adam [24] and lr=10^{-5} for 50K iterations. The VOS module is pre-trained on the YouTubeVOS dataset [47]. We consider two annotation types: *'mask drawing'* and *'corrective clicks'* [5], denoted as Mask and Clicks, respectively. For Clicks, the annotator clicks 3 times to improve the segmentation and decides the number of positive and negative clicks.

Human annotator simulation. We perform experiments only by simulating human intervention. Given $M_{f_*}^t$ and ground-truth mask m_g of f_*, we simulate positive and negative clicks to prompt SAM [26] similar to how a human would. We identify all false-neg and false-pos pixels between m_g and $M_{f_*}^t$. We determine the connected components of each error region, and the center of the largest component is selected as click location, positive or negative.

4. Experimental results

This section presents our experimental results. We first evaluate the frame selection and annotation selection (Sec. 4.2) individually and display the results in Fig. 3(a) and (b), respectively. Finally, we analyze the results of our full pipeline (Sec. 4.3) and report results in Fig. 3(c).

4.1. Frame selection evaluation

Here, we evaluate the frame selection (Sec. 2.2). For a fair comparison among all methods, we use only object-mask as an annotation type (Fig. 3(a)).

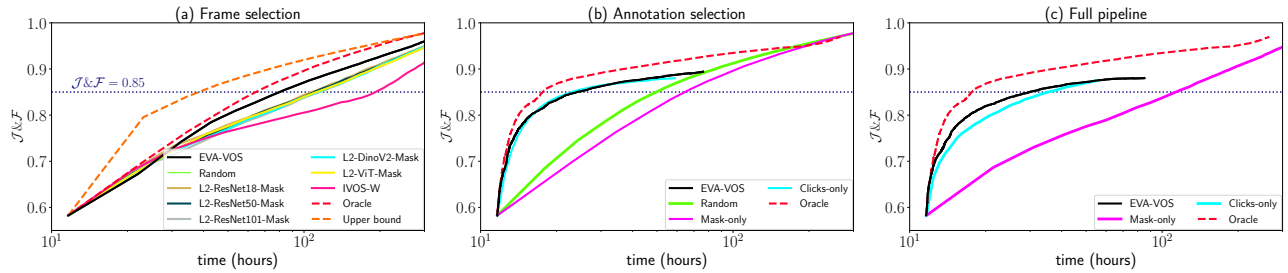

Figure 3: **Experimental results on MOSE.**We report the $\mathcal{J}\&\mathcal{F}$ accuracy as a function of annotation time in hours. (a) The effect of the frame selection stage (for fair comparison we use the same annotation type for all approaches). (b) The effect of the annotation selection stage using the same frame selection (oracle) for all approaches. (c) The results of our full pipeline.

Compared methods. We compare against a baseline that selects frames randomly and against IVOS-W [52] which is the state-of-the-art frame selection method for VOS. For a fair comparison, we train IVOS-W [52] in MOSE-long. We also use powerful image encoders [19, 17, 32] pre-trained for image classification [39] to compute embeddings in Eq. (1) and compare against QNet. We implement an oracle approach that selects the frame with the worst $\mathcal{J}\&\mathcal{F}$ and an upper bound approach that selects the frame with the highest impact on the propagation stage after annotation.

Comparison to the state of the art.

Random is shown as the green line in Fig. 3(a). We run all random baselines 15 times and report the average result.

EVA-VOS (Ours) is shown as the black line in Fig. 3(a)). Given the same annotation time, our framework consistently outperforms Random. For instance, we achieve $\mathcal{J}\&\mathcal{F}$=0.85 at 80.7 hours, 26.7 hours faster than Random.

State-of-the-art frame selection (IVOS-W [52]) performs significantly worse than our method. We observe that our method reaches $\mathcal{J}\&\mathcal{F}$ of 0.85 2.3× faster than IVOS-W.

L2-Encoders yield approximately the same performance as random. This shows that our task-specific QNet learns much better representations and outperforms all pre-trained encoders that have even 28× more parameters.

Oracle is shown as the red dashed line in Fig. 3(a)). Interestingly, we observe that for low budgets (up to 40 hours), our method yields almost identical $\mathcal{J}\&\mathcal{F}$.

Upper bound consistently outperforms the oracle indicating that the frame with the worst $\mathcal{J}\&\mathcal{F}$ is not the most impactful one. Interestingly, the upper bound is only 2.1× faster than our method at $\mathcal{J}\&\mathcal{F} = 0.85$.

4.2. Annotation selection evaluation

Here, we evaluate only our annotation selection stage. For a fair comparison, we set the frame selection for all approaches to oracle, i.e., the frame with the worst $\mathcal{J}\&\mathcal{F}$ is selected to be annotated at each iteration (results in Fig. 3(b)).

Compared methods. We compare our annotation selection to approaches that consider one annotation type (Clicks, Mask), a random approach selecting a_{f_*} randomly, and an oracle that selects the a_{f_*} that yields the maximum quality improvement normalized by the annotation cost.

Comparison to annotation selection methods *EVA-VOS (Ours)* is shown as the black line in Fig. 3(b). It reaches $\mathcal{J}\&\mathcal{F} = 0.85$ in only 29.8 hours.

Random is shown as the green line in Fig. 3(b). Even though it reaches $\mathcal{J}\&\mathcal{F} = 0.9$ at a similar time as our method, it performs significantly worse at lower budgets (e.g., we yield $\mathcal{J}\&\mathcal{F} = 0.85$ 1.9× faster). *Mask-only* performs consistently worse than random at all budgets, indicating that the traditional way of manually drawing object mask [36, 38, 47] is not a good approach.

Clicks-only performs on par with our method at low annotation budgets. However, it plateaus quickly at lower $\mathcal{J}\&\mathcal{F}$ values and it is not able to reach $\mathcal{J}\&\mathcal{F} = 0.9$, whereas our method can yield higher $\mathcal{J}\&\mathcal{F}$ at larger budgets.

Oracle (red dashed line in Fig. 3(b)) performs on par with our method at low budgets. Oracle performs better in very high annotation budgets that reach high $\mathcal{J}\&\mathcal{F}$ above 0.85.

4.3. Frame and Annotation selection evaluation

We evaluate here our full pipeline, showing the effect of both selection modules (results of full pipeline in Fig. 3(c)).

Compared methods. Similar to Sec. 4.2, we compare our method to Clicks-only and Masks-only which select a random frame and consider only one annotation type.

Comparison of annotation methods. *EVA-VOS (Ours)* is shown as the black line in Fig. 3(c) and yield a $\mathcal{J}\&\mathcal{F}$ of 0.85 in 29.8 hours.

Oracle uses both oracle frame selection and annotation selection and shows the trade-off that EVA-VOS could achieve with an ideal oracle training scenario.

Mask-Only resembles the traditional way of annotating videos with object segmentation masks [47, 46, 15, 38, 16]. Our method performs significantly better and achieves a 3.5× speed up compared to Mask-only at $\mathcal{J}\&\mathcal{F} = 0.85$.

Click-Only performs similarly to EVA-VOS at 50 hours but has a worse trade-off for either lower or higher budgets.

Conclusions. We presented the efficient EVA-VOS to annotate objects in videos with segmentation masks. EVA-VOS shows significant gains in annotation time (3.5× speed up) compared to the manual annotation of objects in videos. EVA-VOS reduces the total human annotation time while leading to high-quality segmentation masks.

References

[1] David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. Efficient interactive annotation of segmentation datasets with polygon-rnn++. In *CVPR*, 2018.

[2] Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. Text2live: Text-driven layered image and video editing. In *NeurIPS*, 2022.

[3] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei. What's the point: Semantic segmentation with point supervision. In *ECCV*, 2016.

[4] Maksym Bekuzarov, Ariana Bermudez, Joon-Young Lee, and Hao Li. Xmem++: Production-level video segmentation from few annotated frames. In *ICCV*, 2023.

[5] Rodrigo Benenson, Stefan Popov, and Vittorio Ferrari. Large-scale interactive object segmentation with human annotators. In *CVPR*, 2019.

[6] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *CVPR*, 2017.

[7] Sergi Caelles, Alberto Montes, Kevis-Kokitsi Maninis, Yuhua Chen, Luc Van Gool, Federico Perazzi, and Jordi Pont-Tuset. The 2018 davis challenge on video object segmentation. *arXiv:1803.00557*, 2018.

[8] L. Castrejon, K. Kundu, R. Urtasun, and S. Fidler. Annotating object instances with a Polygon-RNN. In *CVPR*, 2017.

[9] Bowen Chen, Huan Ling, Xiaohui Zeng, Jun Gao, Ziyue Xu, and Sanja Fidler. Scribblebox: Interactive annotation framework for video object segmentation. In *ECCV*, 2020.

[10] Yuhua Chen, Jordi Pont-Tuset, Alberto Montes, and Luc Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *CVPR*, 2018.

[11] Ho Kei Cheng and Alexander G Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *ECCV*, 2022.

[12] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion. In *CVPR*, 2021.

[13] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. In *NeurIPS*, 2021.

[14] Kenan Dai, Jie Zhao, Lijun Wang, Dong Wang, Jianhua Li, Huchuan Lu, Xuesheng Qian, and Xiaoyun Yang. Video annotation for visual tracking via selection and refinement. In *ICCV*, 2021.

[15] Ahmad Darkhalil, Dandan Shan, Bin Zhu, Jian Ma, Amlan Kar, Richard Higgins, Sanja Fidler, David Fouhey, and Dima Damen. Epic-kitchens visor benchmark: Video segmentations and object relations. In *NeurIPS*, 2022.

[16] Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, Philip HS Torr, and Song Bai. MOSE: A new dataset for video object segmentation in complex scenes. In *ICCV*, 2023.

[17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

[18] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[20] Yuk Heo, Yeong Jun Koh, and Chang-Su Kim. Guided interactive video object segmentation using reliability-based attention maps. In *CVPR*, 2021.

[21] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G Schwing. Videomatch: Matching based video object segmentation. In *ECCV*, 2018.

[22] Varun Jampani, Raghudeep Gadde, and Peter V Gehler. Video propagation networks. In *CVPR*, 2017.

[23] Yoni Kasten, Dolev Ofri, Oliver Wang, and Tali Dekel. Layered neural atlases for consistent video editing. 2021.

[24] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2017.

[25] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *CVPR*, 2019.

[26] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[27] Zhuwen Li, Qifeng Chen, and Vladlen Koltun. Interactive image segmentation with latent diversity. In *CVPR*, 2018.

[28] 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 *ECCV*, 2014.

[29] Zhihui Lin, Tianyu Yang, Maomao Li, Ziyu Wang, Chun Yuan, Wenhao Jiang, and Wei Liu. Swem: Towards real-time video object segmentation with sequential weighted expectation-maximization. In *CVPR*, 2022.

[30] Huan Ling, Jun Gao, Amlan Kar, Wenzheng Chen, and Sanja Fidler. Fast interactive object annotation with curve-gcn. In *CVPR*, 2019.

[31] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, 2019.

[32] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

[33] Dim P Papadopoulos, Jasper RR Uijlings, Frank Keller, and Vittorio Ferrari. Extreme clicking for efficient object annotation. In *ICCV*, 2017.

[34] Dim P Papadopoulos, Jasper RR Uijlings, Frank Keller, and Vittorio Ferrari. Training object class detectors with click supervision. In *CVPR*, 2017.

[35] Dim P Papadopoulos, Ethan Weber, and Antonio Torralba. Scaling up instance annotation via label propagation. In *ICCV*, 2021.

[36] 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 *CVPR*, 2016.

[37] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

[38] Jiyang Qi, Yan Gao, Yao Hu, Xinggang Wang, Xiaoyu Liu, Xiang Bai, Serge Belongie, Alan Yuille, Philip HS Torr, and Song Bai. Occluded video instance segmentation: A benchmark. *IJCV*, 2022.

[39] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei. ImageNet large scale visual recognition challenge. *IJCV*, 2015.

[40] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. In *PMLR*, 2017.

[41] Pavel Tokmakov, Jie Li, and Adrien Gaidon. Breaking the" object" in video object segmentation. *arXiv preprint arXiv:2212.06200*, 2022.

[42] Carles Ventura, Miriam Bellver, Andreu Girbau, Amaia Salvador, Ferran Marques, and Xavier Giro-i Nieto. Rvos: End-to-end recurrent network for video object segmentation. In *CVPR*, 2019.

[43] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen. Feelvos: Fast end-to-end embedding learning for video object segmentation. In *CVPR*, 2019.

[44] Ting-Chun Wang, Ming-Yu Liu, Andrew Tao, Guilin Liu, Jan Kautz, and Bryan Catanzaro. Few-shot video-to-video synthesis. In *NeurIPS*, 2019.

[45] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *NeurIPS*, 2018.

[46] Weiyao Wang, Matt Feiszli, Heng Wang, and Du Tran. Unidentified video objects: A benchmark for dense, open-world segmentation. In *ICCV*, 2021.

[47] 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 *ECCV*, 2018.

[48] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K Katsaggelos. Efficient video object segmentation via network modulation. In *CVPR*, 2018.

[49] Zongxin Yang, Yunchao Wei, and Yi Yang. Associating objects with transformers for video object segmentation. *NeurIPS*, 2021.

[50] Zongxin Yang and Yi Yang. Decoupling features in hierarchical propagation for video object segmentation. *NeurIPS*, 2022.

[51] Vickie Ye, Zhengqi Li, Richard Tucker, Angjoo Kanazawa, and Noah Snavely. Deformable sprites for unsupervised video decomposition, 2022.

[52] Zhaoyuan Yin, Jia Zheng, Weixin Luo, Shenhan Qian, Hanling Zhang, and Shenghua Gao. Learning to recommend frame for interactive video object segmentation in the wild. In *CVPR*, 2021.

[53] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ADE20K dataset. In *CVPR*, 2017.