Supplementary Material

READMem: Robust Embedding Association for a Diverse Memory in Unconstrained Video Object Segmentation

Stéphane Vujasinović[1]
stephane.vujasinovic@iosb.fraunhofer.de

Sebastian Bullinger[1]
sebastian.bullinger@iosb.fraunhofer.de

Stefan Becker[1]
stefan.becker@iosb.fraunhofer.de

Norbert Scherer-Negenborn[1]
norbert.scherer-negenborn@iosb.fraunhofer.de

Michael Arens[1]
michael.arens@iosb.fraunhofer.de

Rainer Stiefelhagen[2]
rainer.stiefelhagen@kit.edu

[1] Fraunhofer IOSB*
Ettlingen, Germany

[2] Karlsruhe Institut of Technology
Karlsruhe, Germany

In this supplementary document, we provide additional experiments, visualizations and insights.

A Additional Qualitative Results on the Long-time Video (LV1 [8]) Dataset

We display qualitative results for the READMem variations of MiVOS [2], STCN [3] and QDMN [9] along with their baseline on the LV1 [8] dataset. We use the same settings as described in Section 4 (refer to quantitative results). We also provide the results for XMem [1], which represents the state-of-the-art.

Figures S1, S2 and S3 displays the results for the *blueboy*, *dressage* and *rat* sequences in LV1 [8] respectively when using $s_r = 10$, while Figures S4, S5 and S6 display the results for $s_r = 1$. The estimated segmentation mask of the baselines (MiVOS [2], STCN [3], and QDMN [9]) are visualized in red, while the results of the READMem-based variations (READMem with a baseline) are highlighted in blue. The intersection between the prediction of a baseline and its corresponding READMem variation is depicted in turquoise. The ground-truth contours are highlighted in yellow. We depict XMem [1] results in purple.

*Fraunhofer IOSB is a member of the Fraunhofer Center for Machine Learning.

A.1 Qualitative Results on LV1 [8] with $s_r = 10$

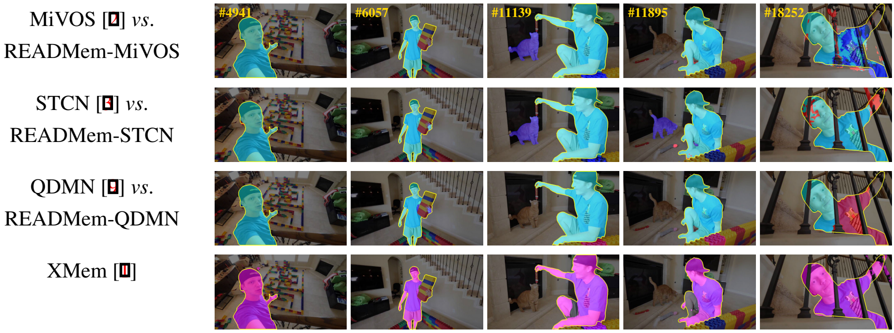

MiVOS [2] *vs.* READMem-MiVOS
STCN [3] *vs.* READMem-STCN
QDMN [9] *vs.* READMem-QDMN
XMem [1]
#4941
#6057
#11139
#11895
#18252

Figure S1: Results on the *blueboy* sequence of LV1 [8] with $s_r = 10$. We depict the results of: the baselines in red, the READMem variations in blue, the intersection between both in turquoise, the ground-truth contours in yellow and XMem [1] results in purple.

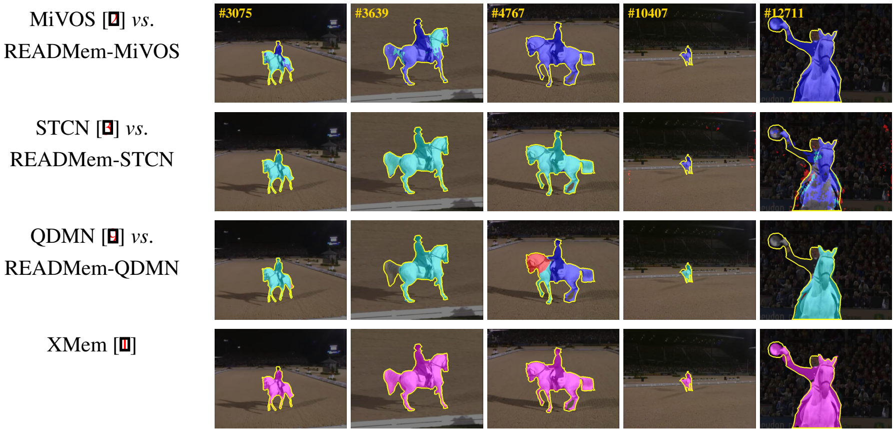

MiVOS [2] *vs.* READMem-MiVOS
STCN [3] *vs.* READMem-STCN
QDMN [9] *vs.* READMem-QDMN
XMem [1]
#3075
#3639
#4767
#10407
#12711

Figure S2: Results on the *dressage* sequence of LV1 [8] with $s_r = 10$. We depict the results of: the baselines in red, the READMem variations in blue, the intersection between both in turquoise, the ground-truth contours in yellow and XMem [1] results in purple.

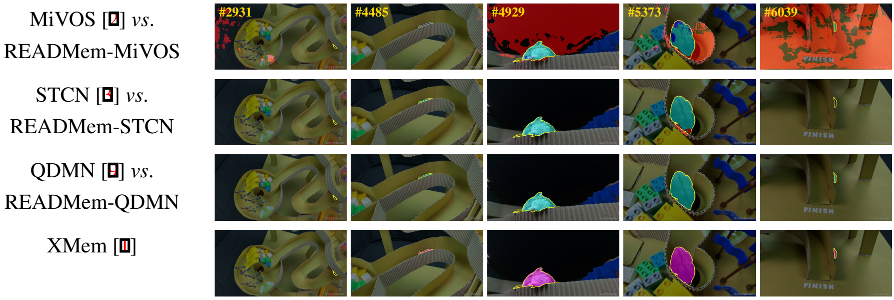

MiVOS [2] *vs.* READMem-MiVOS
STCN [3] *vs.* READMem-STCN
QDMN [9] *vs.* READMem-QDMN
XMem [1]
#2931
#4485
#4929
#5373
#6039

Figure S3: Results on the *rat* sequence of LV1 [8] with $s_r = 10$. We depict the results of: the baselines in red, the READMem variations in blue, the intersection between both in turquoise, the ground-truth contours in yellow and XMem [1] results in purple.

A.2 Qualitative Results on LV1 [8] with $s_r = 1$

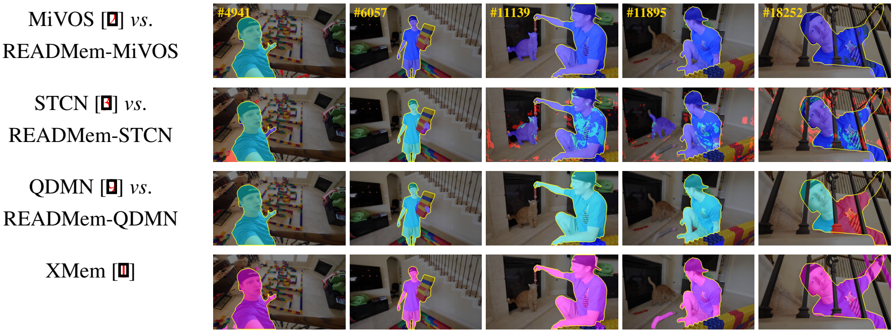

MiVOS [2] *vs.* READMem-MiVOS
STCN [3] *vs.* READMem-STCN
QDMN [9] *vs.* READMem-QDMN
XMem [1]
#4941
#6057
#11139
#11895
#18252

Figure S4: Results on the *blueboy* sequence of LV1 [8] with $s_r = 1$. We depict the results of: the baselines in red, the READMem variations in blue, the intersection between both in turquoise, the ground-truth contours in yellow and XMem [1] results in purple.

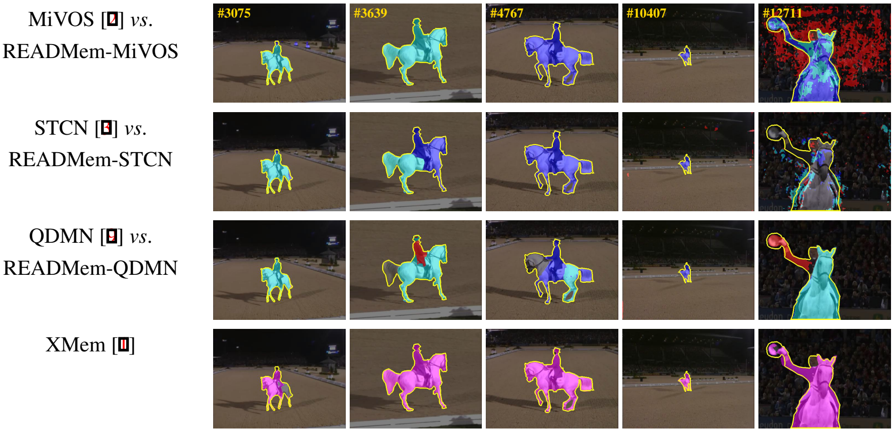

MiVOS [2] *vs.* READMem-MiVOS
STCN [3] *vs.* READMem-STCN
QDMN [9] *vs.* READMem-QDMN
XMem [1]
#3075
#3639
#4767
#10407
#12711

Figure S5: Results on the *dressage* sequence of LV1 [8] with $s_r = 1$. We depict the results of: the baselines in red, the READMem variations in blue, the intersection between both in turquoise, the ground-truth contours in yellow and XMem [1] results in purple.

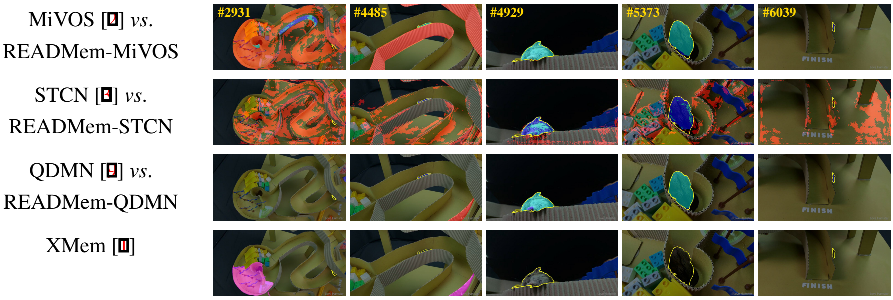

MiVOS [2] *vs.* READMem-MiVOS
STCN [3] *vs.* READMem-STCN
QDMN [9] *vs.* READMem-QDMN
XMem [1]
#2931
#4485
#4929
#5373
#6039

Figure S6: Results on the *rat* sequence of LV1 [8] with $s_r = 1$. We depict the results of: the baselines in red, the READMem variations in blue, the intersection between both in turquoise, the ground-truth contours in yellow and XMem [1] results in purple.

Dataset	# of sequences	avg. length	median length	std.	min. length	max. length
D17 [11] (validation set)	30	67	67	21	34	104
YVOS [12] (validation set)	507	134	144	42	16	180
LV1 [8]	3	2470	2406	1088	1416	3589
VOT2022 [6]	62	321	242	295	41	1500
VOTS2023 [7]	144	2073	1810	1856	63	10699

Table S1: Statistics of popular sVOS and VOT datasets. For more details refer to the original publications.

B Additional Quantitative Evaluation

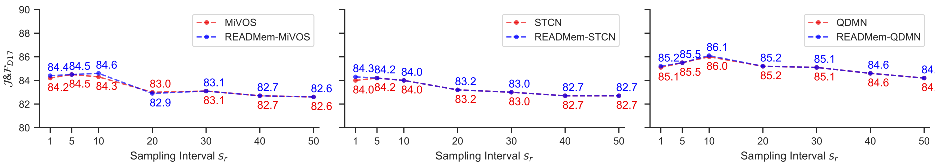
We present in Table S1 useful statistics for popular (sVOS) [8, 11, 12] and Visual Object Tracking (VOT) [6, 7] datasets. As our goal is to allow sVOS methods to perform on long video sequence, Table S1 reveals that the LV1 dataset [8] and the recently introduced VOTS2023 dataset [7] are ideal candidates for assessing the effectiveness of our READMem extension.

Hence, in the main paper we focus on the LV1 dataset [8], to allow for a direct comparison with contemporary sVOS methods. We include the D17 dataset [11] in our evaluations to encompass scenarios with shorter sequences. However, to demonstrate the scalability and versatility of our approach, we also report complementary experiments on VOTS2023 in Table S2. We want to clarify that our method is originally designed for managing the memory of sVOS task, and as such is not modifying the underlying architecture of the sVOS baselines [2, 3, 9], which are not tailored towards handling specific challenges found only in VOT datasets (*e.g.*, small object-to-image ratio, presence of numerous distractors).

B.1 Performance on the DAVIS (D17 [11]) Dataset

We display the performance of MiVOS [2], STCN [3] and QDMN [9] with and without the READMem extension when varying the sampling interval s_r on the D17 [11] dataset, using the same configuration as in Section 4.

In Figure 2 of Section 1, we observe that increasing the sampling interval generally improves the performance of all methods on long videos, regardless of the baseline employed. However, this trend does not hold when working with short video sequences, as shown in Figure S7. Here, we notice a degradation in performance for all methods when using larger sampling intervals.



(a) MiVOS *vs*. READMem-MiVOS (b) STCN *vs*. READMem-STCN (c) QDMN *vs*. READMem-QDMN

Figure S7: Performance comparison of sVOS baselines (MiVOS [2], STCN [3], QDMN [9]) with and without the READMem extension on the D17 [11] dataset, while varying the sampling interval s_r. Regardless of the final performance, we observe a general tendency where increasing the sampling interval (*i.e.*, s_r higher than 10) on short video sequences leads to a performance drop.

Therefore, it is essential to utilize a sampling interval that does not negatively impact the performance on both long and short video sequences. This is where our READMem extension becomes valuable, as it enables the sVOS pipeline to use a small sampling interval (typically $s_r \in [1-10]$) that achieves and maintains high performance for both long and short video sequences.

B.2 Performance as a Function of Memory Size

We explore the impact of the size of the memory on the performance of MiVOS [2], STCN [3] and QDMN [9] with and without our READMem extension on the LV1 [8] dataset. We follow the same experimental setup as in Section 4 (with $s_r = 10$), except for the varying memory size N, which ranges from 5 to 50.

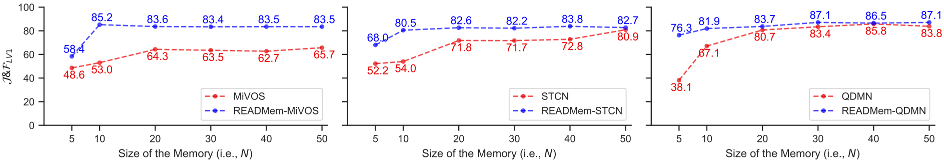
From Figure S8, we observe that the performance of the baselines improves as the memory size increases. Similarly, although to a lesser extent, the READMem variants also demonstrate improved performance with larger memory sizes. However, the READMem variations consistently outperform their respective baselines, especially when using a smaller memory size. This is desired as a smaller memory requires less GPU resources.

Comparing Figure S8 with Figure 2, we notice that increasing the sampling interval (*i.e.*, s_r) of the baselines leads to a significant boost in performance compared to increasing the memory size (*i.e.*, N). Hence, storing a diverse set of embeddings in the memory is more beneficial than including additional ones.

B.3 Performance on the VOTS2023 [7] Dataset

In our quantitative evaluation (refer to Table 1 of Section 4), we demonstrate and analyze the effectiveness of our approach on sVOS datasets, encompassing both short (*i.e.*, D17 [11]) and long (*i.e.*, LV1 [8]) sequences, to allow for a direct comparison with contemporary sVOS approaches (*i.e.*, [1, 8]). In an effort, to enhance the soundness of our READMem extension, we conduct additional experiments on the VOTS2023 dataset [7]. We tabulate in Table S2, the results of sVOS baselines [2, 3, 9] with and without READMem on the VOTS2023 tracking benchmark.

For the evaluation we use the same settings as described in Section 4 (refer to quantitative results) and the official VOT evaluation toolkit (version 0.6.4 released on the 31 May 2023 – `https://github.com/votchallenge/toolkit`). We observe from Table S2, that the READMem variants consistently outperform their baseline counterpart.



(a) MiVOS *vs*. READMem-MiVOS (b) STCN *vs*. READMem-STCN (c) QDMN *vs*. READMem-QDMN

Figure S8: We compare the performance of sVOS baselines (MiVOS [2], STCN [3], QDMN [9]) with and without the READMem extension on the LV1 [8] dataset while varying the size of the memory (*i.e.*, N). A general tendency is that increasing the memory size, leads to better performance.

		(Higher is better)			(Lower is better)	
Method	**Q**	**ACC**	**ROB**	**ADQ**	**NRE**	**DRE**
MiVOS [2] (CVPR 21)	0.38	0.55	0.54	0.75	0.41	0.06
MiVOS [2] ($s_r = 50$) (CVPR 21)	$0.39^{\uparrow 0.01}$	0.55	$0.58^{\uparrow 0.04}$	$0.67^{\downarrow 0.08}$	$0.35^{\downarrow 0.06}$	$0.07^{\uparrow 0.01}$
READMem-MiVOS (ours)	$0.43^{\uparrow 0.05}$	$0.57^{\uparrow 0.02}$	$0.60^{\uparrow 0.06}$	$0.67^{\downarrow 0.08}$	$0.33^{\downarrow 0.08}$	0.06
STCN [3] (NIPS 21)	0.40	0.55	0.62	0.67	0.29	0.08
STCN [3] ($s_r = 50$) (NIPS 21)	0.40	0.55	$0.61^{\downarrow 0.01}$	$0.61^{\downarrow 0.06}$	0.29	$0.10^{\uparrow 0.02}$
READMem-STCN (ours)	$0.42^{\uparrow 0.02}$	$0.56^{\uparrow 0.01}$	$0.66^{\uparrow 0.04}$	$0.57^{\downarrow 0.10}$	$0.25^{\downarrow 0.04}$	$0.09^{\uparrow 0.01}$
QDMN [9] (ECCV 22)	0.44	0.59	0.62	0.69	0.28	0.10
QDMN [9] ($s_r = 50$) (ECCV 22)	$0.42^{\downarrow 0.02}$	0.59	$0.60^{\downarrow 0.02}$	$0.63^{\downarrow 0.06}$	$0.30^{\uparrow 0.02}$	$0.11^{\uparrow 0.01}$
READMem-QDMN (ours)	$0.45^{\uparrow 0.01}$	0.59	$0.63^{\uparrow 0.01}$	$0.67^{\downarrow 0.02}$	$0.27^{\downarrow 0.01}$	$0.09^{\downarrow 0.01}$

Table S2: Quantitative evaluation of sVOS methods [1, 2, 3, 9] with and without READMem on the VOTS2023 [7] datasets. We use the same settings as described in Section 4 and the official VOT evaluation toolkit.

In contrast to previous VOT challenges [4, 5, 6], VOTS2023 introduced new evaluation metrics split into: (i) a primary performance metric: The Tracking Quality (**Q**) and (ii) secondary metrics: the Accuracy (**ACC**), Robustness (**ROB**), Not-Reported Error (**NRE**), Drift-Rate Error (**DRE**) and Absence-Detection Quality (**ADQ**). Please refer to the VOTS2023 paper for more details.

C Initialization of the Memory

We investigate the performance variation when employing two different initialization for READMem in Table S3: The strategies are as follows: (1) integrates every t-th frame into the memory until full, while (2) fills the memory slots with the embeddings of the annotated frame and includes a new frame to the memory if the conditions on the lower bound on similarity and the Gramian are met (follows a greedy approach). The second strategy yields worse results on the short scenarios and is slightly below the performance of strategy (1) on LV1 [8]. We argue that with longer sequences the memory has more opportunities to integrate decisive frame representations in the memory to use as a reference. Hence, initialization plays a crucial role in short videos, but as the method observes longer videos and has access to a larger pool of frames to select from, the importance diminishes.

D Discussion and Limitations

We are aware of the limitations imposed by the hand-crafted threshold for the lower similarity bound l_{sb}, although to avoid any fine-tuning, we set the threshold value to 0.5. A

	READMem-MiVOS		**READMem-STCN**		**READMem-QDMN**	
Initialization	$\mathcal{J}\&\mathcal{F}_{\text{LV1}}$	$\mathcal{J}\&\mathcal{F}_{\text{D17}}$	$\mathcal{J}\&\mathcal{F}_{\text{LV1}}$	$\mathcal{J}\&\mathcal{F}_{\text{D17}}$	$\mathcal{J}\&\mathcal{F}_{\text{LV1}}$	$\mathcal{J}\&\mathcal{F}_{\text{D17}}$
(1)	83.6	84.6	82.6	84.0	84.0	86.1
(2)	82.7	73.7	85.3	73.6	72.5	73.3

Table S3: Performance variation when leveraging two different initialization strategies for READMem-MiVOS. Besides the initialization strategy, the remaining parameters are consistent to Section 4 (we set $s_r = 10$).

more thoughtful approach would incorporate a learnable parameter. This approach could potentially lead to improved performance, albeit at the expense of the plug-and-play nature of our extension. Another point for improvement is to reduce the participation of the background when computing the similarity between two embeddings. A possible enhancement is to integrate either the segmentation mask estimated by the sVOS pipeline or use the memory values to estimate a filter that can be applied to the memory keys before computing a similarity score.

E Training

For our experiments, we utilize the original weights provided by the authors of MiVOS [2], STCN [3], and QDMN [9]. Our primary focus is to showcase the benefits of our extension (*i.e.*, READMem) without modifying the baselines. To provide a comprehensive overview of the baselines, we briefly elaborate on the training methodology. The training procedure follows the regiment presented in STM [10] and refined in the subsequent work, MiVOS [2]. The training is divided into two stages employing the bootstrapped cross-entropy loss [2] and utilizing the Adam optimizer (refer to the original papers [2, 3, 9] and their supplementary materials for detailed insights).

The training comprises the following stages: (1) A pre-training stage, in which static image datasets are used as in [10] to simulate videos consisting of three frames. While all three frames originate from the same image, the second and third frames are modified using random affine transformations (2) A main-training stage, which uses the DAVIS [11] and the Youtube-VOS [12] datasets (which provide real videos). Similar to the pre-training stage, three frames from a video are sampled, gradually increasing the temporal gap from 5 to 25 frames during training. Subsequently, the temporal gap is annealed back to 5 frames, following a curriculum training approach [13]. (Optional) Moreover, after the pre-training stage a synthetic dataset BL30K [2] can be leveraged to enhance the ability of the model to better handle complex occlusion patterns.

References

[1] Ho Kei Cheng and Alexander G. Schwing. XMem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *European Conference on Computer Vision (ECCV)*, 2022.

[2] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

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

[4] Matej Kristan et al. The eighth visual object tracking vot2020 challenge results. In *European Conference on Computer Vision (ECCV)*, 2020.

[5] Matej Kristan et al. The ninth visual object tracking vot2021 challenge results. In *International Conference on Computer Vision (ICCV)*, 2021.

[6] Matej Kristan et al. The tenth visual object tracking vot2022 challenge results. In *European Conference on Computer Vision (ECCV)*, 2022.

[7] Matej Kristan et al. The vots2023 challenge performance measures. 2023.

[8] Yongqing Liang, Xin Li, Navid Jafari, and Jim Chen. Video object segmentation with adaptive feature bank and uncertain-region refinement. In *Neural Information Processing Systems (NeurIPS)*, 2020.

[9] Yong Liu, Ran Yu, Fei Yin, Xinyuan Zhao, Wei Zhao, Weihao Xia, and Yujiu Yang. Learning quality-aware dynamic memory for video object segmentation. In *European Conference on Computer Vision (ECCV)*, 2022.

[10] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *International Conference on Computer Vision (ICCV)*, 2019.

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

[12] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018.

[13] Peng Zhang, Li Hu, Bang Zhang, Pan Pan, and DAMO Alibaba. Spatial consistent memory network for semi-supervised video object segmentation. In *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020.