Supplementary Material
Scalable Video Object Segmentation with Simplified Framework

Qiangqiang Wu[1] Tianyu Yang[2*] Wei Wu[1] Antoni B. Chan[1]
[1]Department of Computer Science, City University of Hong Kong
[2]International Digital Economy Academy
{qiangqwu2-c, weiwu56-c-c}@my.cityu.edu.hk, tianyu-yang@outlook.com
abchan@cityu.edu.hk

Config	Value
optimizer	AdamW [6]
base learning rate	2e-5
weight decay	1e-7
droppath rate	0.1
batch size	32
Iterations	210,000
learning rate decay iteration	125,000
learning rate schedule	steplr
maximum sampling frame gap	10
training set	DAVIS [10] + YT-VOS [12]

Table 1: The training parameters of SimVOS used for DAVIS [9, 10] evaluation.

Config	Value
optimizer	AdamW [6]
base learning rate	2e-5
weight decay	1e-7
droppath rate	0.25
batch size	32
Iterations	210,000
learning rate decay iteration	125,000
learning rate schedule	steplr
maximum sampling frame gap	15
training set	YT-VOS [12]

Table 2: The training parameters of SimVOS used for YouTube-VOS 19 [12] evaluation.

In this supplementary material, we provide detailed implementation details, additional ablation study, completed comparison on YouTube-VOS 19, and more qualitative and quantitative results to demonstrate the effectiveness of the proposed *Simplified VOS framework* (SimVOS). Specifically, Sec. A shows the detailed training details and architectures for our SimVOS. Sec. C shows the speed comparison on V100 platform. More completed quantitative and qualitative results are respectively presented in Sec. B and Sec. D.

A. Implementation Details

Training hyperparameters. The training details of our SimVOS are shown in Tables 1 and 2. Following the previous VOS approaches [8, 14, 13], different training data sources are used to train our SimVOS model, which depends on the target evaluation benchmark. Specifically, for DAVIS 16/17 [9, 10] evalution, the combination of the training splits in both DAVIS-17 [10] and YouTube-VOS 19 [12] is used for training. For YouTube-VOS 19 evaluation, only the training split in its own dataset is used. During the training stage, only a pair of frames is randomly sampled within the predefined maximum sampling frame gap. We use a larger maximum sampling frame gap (i.e., 15) for the YouTube VOS evaluation since the videos in this dataset are commonly longer than the videos in the DAVIS datasets. To alleviate overfitting and generalize well to unseen objects in YouTube-VOS, a larger droppath rate (i.e., 0.25) is employed for training.

Architecture of the token refinement (TR) module. The TR module consists of a convolutional layer and a fully-connected layer, which is denoted as $f(\cdot)$ in Eq. 3 of the main paper. The convolutional layer firstly reduces the input channel of $(C+1)$ to $C/4$ with a 3×3 kernel, and the output is activated with a GELU [5] function. The fully-connected layer further maps the input channel of $c/4$ to K for the following prototype generation, which is detail in Fig. 4 of the main paper.

Training. The training is conducted on 8 NVIDIA A100 GPUs, which takes about 15 hours to finish the whole main training on video datasets.

[*] Corresponding Author

Method	S	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}_{seen}\uparrow$	$\mathcal{F}_{seen}\uparrow$	$\mathcal{J}_{unseen}\uparrow$	$\mathcal{F}_{unseen}\uparrow$
MiVOS* [1]	✓	82.4	80.6	84.7	78.1	86.4
HMMN [11]	✓	82.5	81.7	86.1	77.3	85.0
STCN [3]	✓	82.7	81.1	85.4	78.2	85.9
STCN* [3]	✓	84.2	82.6	87.0	79.4	87.7
SwinB-AOT [14]	✓	84.5	84.0	88.8	78.4	86.7
XMEM [2]	✓	85.5	84.3	88.6	80.3	88.6
XMEM* [2]	✓	85.8	84.8	89.2	80.3	88.8
CFBI [13]		81.0	80.6	85.1	75.2	83.0
CFBI+ [15]		82.6	81.7	86.2	77.1	85.2
JOINT [7]		82.8	80.8	84.8	79.0	86.6
SSTVOS [4]		81.8	80.9	-	76.7	-
XMEM^{-} [2]		**84.2**	**83.8**	**88.3**	78.1	86.7
SimVOS-BS		82.2	81.7	86.1	76.4	84.7
SimVOS-B		**84.2**	83.1	87.5	**79.1**	**87.2**

Table 3: Comparisons with previous approaches on the YouTube-VOS 2019 validation set. S indicates the usage of synthetic data pre-training. $*$ denotes the BL30K [1] pretraining. $-$ means without applying synthetic data pre-training. We use the default 480P 6 FPS videos evaluation on YouTube-VOS 2019.

Method	DAVIS-16			DAVIS-17			YT-19		
	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	FPS	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	FPS	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}_{unseen}$	FPS
STCN (NeurIPS'21)	-	-	26.9	82.5	79.3	20.2	-	-	13.2
SwinB-DeAOT-L (NeurIPS'22)	89.8	88.7	-	83.8	81.0	15.4	82.0	76.1	11.9
XMEM (ECCV'22)	90.8	89.6	29.6	84.5	81.4	22.6	**84.2**	78.1	22.6
SimVOS-BS	91.5	89.9	12.3	87.1	84.1	8.0	82.2	76.4	7.5
SimVOS-B	**92.9**	**91.3**	7.2	**88.0**	**85.0**	3.5	**84.2**	**79.1**	3.3

Table 4: Performance and FPS comparison between our SimVOS and SOTA approaches. All methods use a **single training stage** on DAVIS17+YT-19) for fair comparison. FPS is measured on one V100.

B. Results on YouTube-VOS 19

We show the complete results on YouTube-VOS 19 [12] in Table 3. Our methods perform favorably against state-of-the-art VOS approaches under the same training setting. Specifically, SimVOS-B achieves better performance on unseen objects than the other approaches, which shows its generalization ability to new objects. Although our efficient variant (SimVOS-BS) obtains inferior results to SimVOS-B, it still outperforms the other transformer-based approach (SSTVOS) in terms of the $\mathcal{J}\&\mathcal{F}$ metric.

C. Speed Comparison on the V100 platform

Tab. 4 shows the speed using a V100 on 3 datasets. Despite its lower speed, our SimVOS-B gets best performance on 3 VOS benchmarks w/ the naive memory mechanism, which demonstrates its strong matching ability. Our TR module reduces the generated tokens to speed-up VOS. Other solutions are also possible, e.g., modifying ViT to be more efficient. We leave this as future work since our aim is to bridge the gap between VOS and self-supervised pre-training ViT communities, inspiring future works in VOS pre-training.

D. Qualitative Visualization

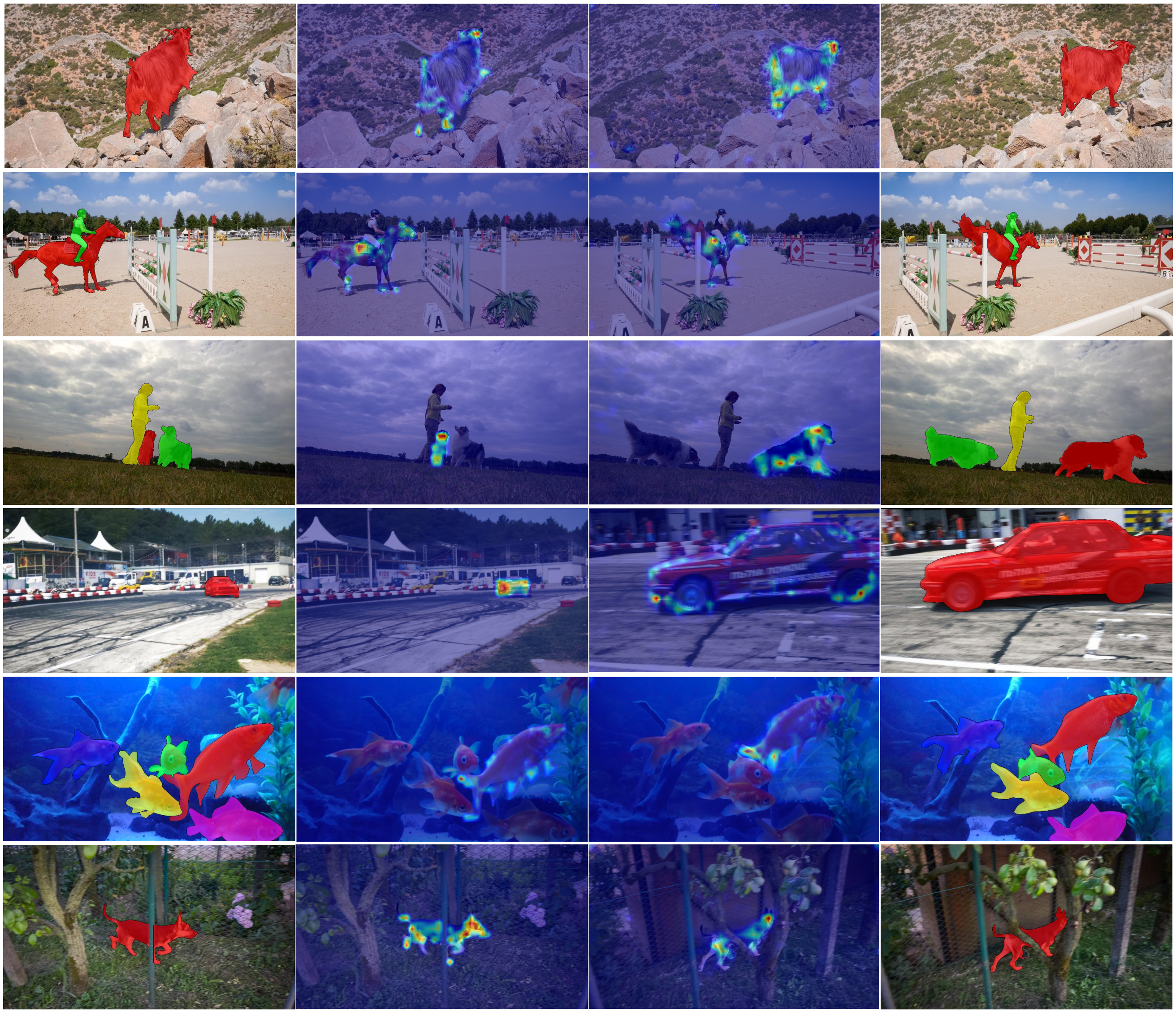
We show more qualitative visualization in Fig. 1. The visualization of the attention in foreground prototype generation indicates that the TR module tends to aggregate discriminative boundary features. This can be explained that the local boundary cues play an essential role in accurate VOS.

References

[1] H.K. Cheng, Y.W. Tau, and C.K. Tang. Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion. In *CVPR*, 2021.

[2] H. K. Cheng and A G. Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *ECCV*, 2022.

[3] H. K. Cheng, Y. W. Tai, and C. K. Tang. Rethinking space-time networks with improved memory coverage for efficient



(a) 1-st Frame w/ Mask Annotation

(b) Fore. Prototype Generation at the 1-st Frame

(c) Fore. Prototype Generation at t-th Frame

(d) Mask Predicted at $(t+1)-\mathrm{th}$ Frame

Figure 1: Visualization of (a) the mask annotation in the first frame, foreground prototype generation in both the (a) first frame and (b) previous frame (i.e., t-th frame), and the mask prediction in the following $(t+1)$-th frame. If multiple objects exist, the object annotated with red mask is chosen for visualization of the foreground prototype generation.

video object segmentation. In *NIPS*, pages 11781–11794, 2021.

[4] B. Duke, A. Ahmed, C. Wolf, and G. W. Taylor. Sstvos: Sparse spatiotemporal transformers for video object segmentation. In *CVPR*, pages 5912–5921, 2021.

[5] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). In *arXiv:1606.08415*, 2016.

[6] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *arXiv:1412.6980*, 2014.

[7] Y. Mao, N. Wang, W. Zhao, and H. Li. Joint inductive and transductive learning for video object segmentation. In *ICCV*, 2021.

[8] S.W. Oh, J.Y. Lee, N. Xu, and S.J. Kim. Video object segmentation using space-time memory networks. In *ICCV*, 2019.

[9] 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.

[10] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool. The 2017 davis challenge on video object segmentation. In *arXiv:1704.00675*, 2017.

[11] H. Seeing, S.W. Oh, J.Y. Lee, S. Lee, S. Lee, and E. Kim. Hierarchical memory matching network for video object seg-

mentation. In *ICCV*, 2021.

[12] N. Xu, L. Yang, Y. Fan, D. Yue, Y. Liang, J. Yang, and T. Huang. Youtube-vos: A large-scale video object segmentation benchmark. In *arXiv:1809.03327*, 2018.

[13] Z. Yang, Y. Wei, and Y. Yang. Collaborative video object segmentation by foreground-background integration. In *ECCV*, 2020.

[14] Z. Yang, Y. Wei, and Y. Yang. Associating objects with transformers for video object segmentation. In *NIPS*, pages 2491–2502, 2021.

[15] Z. Yang, Y. Wei, and Y. Yang. Collaborative video object segmentation by multi-scale foreground-background integration. In *TPAMI*, 2021.