Supplementary Material: Multi-grained Temporal Prototype Learning for Few-shot Video Object Segmentation

Nian Liu[1] Kepan Nan[2] Wangbo Zhao[3] Yuanwei Liu[2] Xiwen Yao[2†]
Salman Khan[1,4] Hisham Cholakkal[1] Rao Muhammad Anwer[1] Junwei Han[2] Fahad Shahbaz Khan[1,5]
[1] Mohamed bin Zayed University of Artificial Intelligence [2] Northwestern Polytechnical University
[3] National University of Singapore
[4]Australian National University [5]CVL, Linköping University

Setting	Mask	Selection	$\mathcal{J}$-Mean	$\mathcal{F}$-Mean	mVC$_7$
w/o Memory	-	-	62.5	60.3	62.1
Upper Bound	GT	Rand.	65.8	64.1	64.4
Lower Bound	Pred.	Rand.	62.2	61.0	62.3
Ours	Pred.	RMS	63.5	61.9	62.8

Table 1. **Necessity of reliable memory selection.** "GT": Using ground truth memory masks $\mathbf{O^m}$ during testing. "Pred.": Using predicted memory masks $\tilde{\mathbf{O}}^{\mathbf{m}}$ during testing. "Rand.": Random selection. "RMS": our reliable memory selection.

1. Necessity of Reliable Memory Selection

We conduct experiments to verify the necessity of using memory and reliable memory information for FSVOS, and the results are illustrated in Table 1. We define an upper bound of the memory information usage by randomly selecting T_m memory frames and using their ground truth masks $\mathbf{O^m}$ for memory prototype learning during testing. Meanwhile, adopting predicted masks $\tilde{\mathbf{O}}^{\mathbf{m}}$ for the randomly selected memory is the lower bound. We find that the upper bound setting achieves significant performance improvement compared with the "w/o Memory" setting, demonstrating the great potential of using memory prototype learning. However, comparing the lower bound design with "w/o Memory", we find that when using predicted masks for memory, simple random selection can not guarantee performance improvement due to noisy masks. This also shows the difference between FSVOS and semi-supervised VOS, in which usually using memory information definitely brings performance gain. This encourages us to design our reliable memory selection method (last row), which effectively improves the model performance. However, we also find that its performance is still largely behind the upper bound design. Hence, better memory usage methods should be explored in future works.

†Corresponding author: yaoxiwen517@gmail.com.

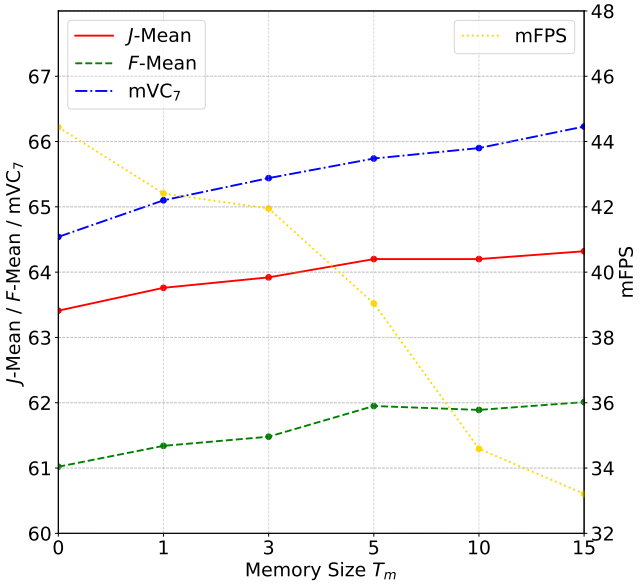

Figure 1. **Performance and speed comparison of using different memory sizes.**

2. Ablation Study on Memory Size

Since we introduce T_m historical frames for providing memory guidance information, we conduct experiments of using different memory size T_m during inference for our VIPMT model, and report results in Figure 1. When using $T_m = 0$ memory frames, we simply remove the memory prototype.

The figure shows that simply introducing more memory frames can bring progressive improvements on the mVC$_7$ metric, which measures temporal prediction consistency among long-range adjacent frames. However, when T_m reaches 5, more temporal information brings little influence on metric $\mathcal{J}$ and even degrades the performance on metric $\mathcal{F}$, which demonstrates that five historical frames can pro-

vide enough memory information for our model while more frames may simply provide temporal redundancies, which is similar to the findings in [5].

Intuitively, more memory frames also cause larger computational costs. Hence, we also provide average FPS on four folds, denoted as "mFPS", of different memory settings in Figure 1. We find that FPS drops gradually when using more frames, especially when we change the memory size from 5 to 10. By comprehensively considering the trade-off between speed and performance, we consider 5 as the best setting and use it in our final model to get good performance as well as fast segmentation speed.

3. More Analysis on the Necessity of Using Structural Similarity Maps

We conduct experiments on our IoUNet to verify the necessity of our proposed structural similarity maps (SSM). We report mean results of mean absolute error (MAE) and accuracy for IoU prediction on the four folds of YouTube-VIS for using SSM or not in Table 2. The accuracy is computed based on Tab. 7 of our paper. We compute the accuracy of predicting each frame's IoU is larger than the threshold or not (*i.e.*, binary classification) and report the average accuracy of varying thresholds (0.5 to 0.9). The results clearly show that using SSM leads to more accurate IoU prediction.

	w/o SSM	**w SSM**
MAE	0.236	0.218
Accuracy	0.686	0.710

Table 2. **Experimental results of MAE and accuracy for IoU prediction.** SSM means the proposed structural similarity maps.

4. Model Runtime Comparison

We report inference speed (on a single A100 GPU) comparison of different methods in Table 3. Besides our baseline IPMT model [6] and two existing FSVOS methods, *i.e.*, [1, 7], we also include the three compared semi-supervised VOS methods in Tab. 3 of our paper for comparison, *i.e.* STCN [3], XMem [2], and RDE-VOS [4]. We can find that our VIPMT is only 9% slower than IPMT and obtains comparative speed compared to TTI.

Model	VIPMT	IPMT	DAN	TTI	IPMT +STCN	IPMT +XMem	IPMT +RDE-VOS
FPS	39.04	43.13	77.53	40.35	66.99	66.48	49.51

Table 3. **Runtime Comparison.**

5. Prototype Distribution Comparison

Since our baseline IPMT is motivated by reducing the gap between support and query distribution (see Fig.1 and Tab. 7 in the IPMT paper [6]), it is expected that VIPMT should have the similar property. We follow Tab. 7 in the IPMT paper and report the distribution gap measured by Euclidean distances among the query, support, and the final frame-level intermediate ($\mathbf{G}_5^{\mathbf{f}_i}$) prototypes on the four folds of YouTube-VIS in Table 4. It shows that the distribution distances after using VIPMT (query-intermediate distance D_{qi} and support-intermediate distance D_{si}) are smaller than those of before using it (query-support distance D_{qs}).

	Fold-1	Fold-2	Fold-3	Fold-4	Mean
D_{qs}	1.600	2.012	0.905	2.077	1.649
D_{si}	1.466	1.579	0.686	1.691	1.356
D_{qi}	1.364	1.088	0.669	0.989	1.028

Table 4. **Intra-class diversity measured by Euclidean distances among the query, support, and intermediate prototypes of four folds on YouTube-VIS.** D_{qs} means the distance between the original query and support. D_{qi} and D_{si} denote the query-intermediate distance and support-intermediate distance after using VIPMT, respectively.

6. More Visual Comparison with State-of-the-art Methods

We give more visual comparison results of our VIPMT model against state-of-the-art methods DAN[1] and TTI[7] in a video file. It shows that our VIPMT can handle many challenging scenarios well, *i.e.*, big objects, fast moving objects, multiple objects, cluttered backgrounds, etc, while DAN and TTI are heavily disturbed in these scenarios.

References

[1] Haoxin Chen, Hanjie Wu, Nanxuan Zhao, Sucheng Ren, and Shengfeng He. Delving deep into many-to-many attention for few-shot video object segmentation. In *CVPR*, pages 14040–14049, 2021. 2

[2] Ho Kei Cheng and Alexander G Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *ECCV*, pages 640–658. Springer, 2022. 2

[3] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. *NeurIPS*, 34:11781–11794, 2021. 2

[4] Mingxing Li, Li Hu, Zhiwei Xiong, Bang Zhang, Pan Pan, and Dong Liu. Recurrent dynamic embedding for video object segmentation. In *CVPR*, pages 1332–1341, 2022. 2

[5] 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*, pages 1362–1372, 2022. 2

[6] Yuanwei Liu, Nian Liu, Xiwen Yao, and Junwei Han. Intermediate prototype mining transformer for few-shot semantic segmentation. In *NeurIPS*, 2022. 2

[7] Mennatullah Siam, Konstantinos G Derpanis, and Richard P Wildes. Temporal transductive inference for few-shot video object segmentation. *arXiv preprint arXiv:2203.14308*, 2022. 2