Supplementary Material: Per-Clip Video Object Segmentation

Kwanyong Park[1] Sanghyun Woo[1] Seoung Wug Oh[2] In So Kweon[1] Joon-Young Lee[2]

[1]KAIST [2]Adobe Research

1. Discussion about Per-Clip Inference

We believe that the main applications of semi-supervised VOS are offline scenarios (*e.g.* video editing) given the requirement of one GT mask input, and our approach can make great improvements in speed and accuracy for such applications.

Online applications might be considered when the VOS method is combined with other techniques (*e.g.* instance segmentation) which initialize the target. Even in this case, our method can process the video in a near-online manner with a shorter clip length. Practical downsides might be i) a few frames delayed output (at most the clip length-1 frames) to perform the clip-level optimization (ICR) and ii) small additional latency by PMM. On the other hand, one major benefit is that even for the case when the frame input rate is faster than the per-frame processing time of the baseline (*i.e.* STCN), our method can process the abundant input frames, thanks to our efficient per-clip approach.

2. Additional Ablation Study and Analysis

Additional Component-wise Ablation. Table 1 shows an extended version of ablation study. Each module shows unique performance improvements. More detailed analyses of each module are in Sec. 4.2 of the main paper.

Analysis on Intra-Clip Refinement (ICR). We perform an ablation study on size of local window in intra-clip refinement module. Specifically, we vary the spatial and temporal window size from our default setting (*i.e.* temporal window size 2 and spatial window size 7). The overall score is reported on the Youtube-VOS [1] 2019 validation set.

Table 2 shows the performance for different spatial window size. We vary the spatial window size from 3 to ∞. If the spatial window size is too small (*e.g.* 3 or 5), it might be hard to capture relevant information from other frames due to the motion of objects, resulting in performance degradation. On the other hand, without any locality constraint (*i.e.* ∞ in Table 2), the ambiguity of correspondence leads to significant performance drop. It implies that imposing the locality constraint is crucial to avoid noisy propagation. For spatial window size between 7 and 11, the model is robust to change of the hyperparameter and shows great performance. Among them, we set spatial window size as 7 due to lower computation cost and slightly better performance.

Table 3 summarizes the performance for different temporal window size. While all candidates obtain strong performance, we pick temporal window size as 2 for saving in computation.

Method	PMM	PCT	ICR	Clip Length (L)			
				L=5	L=10	L=15	L=25
STCN				82.7	81.9	79.6	78.1
	✓			82.7	82.3	81.7	81.1
		✓		83.6	82.6	81.8	80.5
			✓	83.1	82.5	81.6	80.3
		✓	✓	84.6	83.4	82.8	81.4
	✓		✓	83.1	82.6	82.3	81.8
	✓	✓		83.6	83.0	82.5	81.8
Ours	✓	✓	✓	**84.6**	**84.1**	**83.6**	**83.0**

Table 1. **Additional module ablation study.**

Spatial Window	Clip Length (L)			
	L=5	L=10	L=15	L=25
3	83.9	82.7	82.6	81.9
5	84.4	83.6	83.3	82.6
7	**84.6**	84.1	83.6	**83.0**
9	84.3	84.0	83.8	82.8
11	84.5	**84.2**	**83.9**	82.9
15	84.4	83.8	83.5	82.5
∞	80.8	80.0	79.3	77.8

Table 2. **Spatial window size in ICR.**

Temporal Window	Clip Length (L)			
	L=5	L=10	L=15	L=25
2	**84.6**	**84.1**	83.6	83.0
5	84.3	**84.1**	83.8	83.0
10	-	84.0	**83.9**	**83.1**

Table 3. **Temporal window size in ICR.**

Analysis on Progressive Matching Mechanism (PMM).

We study the impact of segment length, which controls the frame interval of temporary memory in PMM. Table 4 shows quantitative results under different segment length F. We choose Ours-L10 as an example. Note that PMM is not used when $F = L$ (*i.e.* F=10). As the segment length decreases, the size of augmented memory in PMM increases and the model becomes inefficient. On the longer segment setting, the augmented size is negligible compared to the main memory and it makes low computational overhead. However, in the shorter setting, time spent in PMM increases near-linearly for the increasing extra memory.

With our default length of segments (*i.e.* F=5), the PMM pushes the performance of longer clip settings significantly (Table 1) while introducing slight overheads.

	Segment Length (F)			
	F=1	F=2	F=**5**	F=10
Overall	83.3	83.9	**84.2**	83.4
FPS	16.9	17.9	**21.8**	22.5

Table 4. **Segment length in PMM.**

References

[1] 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:1809.03327*, 2018. 1