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

Abstract

Recently, memory-based approaches show promising results on semi-supervised video object segmentation. These methods predict object masks frame-by-frame with the help of frequently updated memory of the previous mask. Different from this per-frame inference, we investigate an alternative perspective by treating video object segmentation as clip-wise mask propagation. In this per-clip inference scheme, we update the memory with an interval and simultaneously process a set of consecutive frames (i.e. clip) between the memory updates. The scheme provides two potential benefits: accuracy gain by clip-level optimization and efficiency gain by parallel computation of multiple frames. To this end, we propose a new method tailored for the per-clip inference. Specifically, we first introduce a clip-wise operation to refine the features based on intra-clip correlation. In addition, we employ a progressive matching mechanism for efficient information-passing within a clip. With the synergy of two modules and a newly proposed per-clip based training, our network achieves state-of-the-art performance on Youtube-VOS 2018/2019 val (84.6% and 84.6%) and DAVIS 2016/2017 val (91.9% and 86.1%). Furthermore, our model shows a great speed-accuracy trade-off with varying memory update intervals, which leads to huge flexibility.

1. Introduction

The goal of semi-supervised video object segmentation (VOS) is to segment foreground objects in every frame of a video given a ground truth object mask in the first frame. One of the latest breakthroughs in this task is a memory-based approach that the Space-Time Memory network (STM) [29] proposed. STM encodes and stores the past frames with the corresponding masks as memory (*i.e.* memory update step) then estimates the mask of the current (query) frame through learned spatio-temporal memory matching (*i.e.* mask prediction step). It iterates the mem-
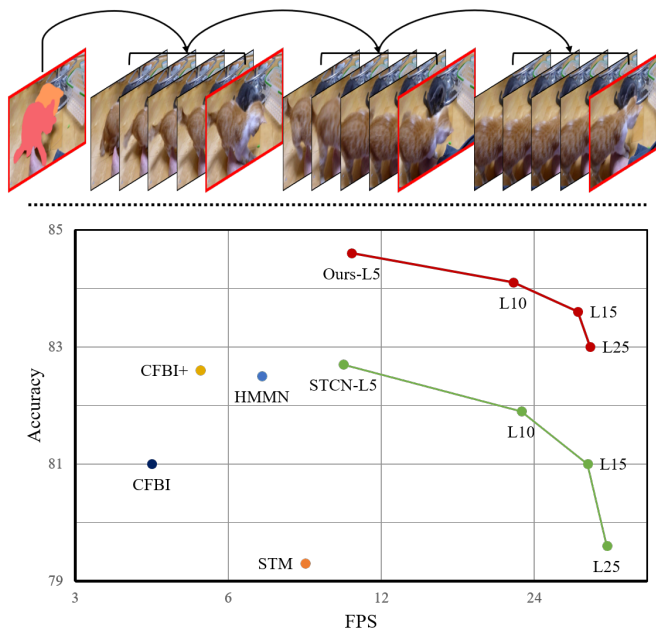

Figure 1. **(top)** An illustrative example of per-clip inference where the memory update interval L is 5. We mark memory frames using red image borders. **(bottom)** Accuracy vs. FPS - We compare our model under the different inference setting of L with SOTA methods [9, 29, 35, 50, 52]. We report the overall score and FPS on Youtube-VOS 2019 [47] validation set. For a fair comparison, we compute the FPS of all the reported methods using the same machine. We additionally report STCN variants, extended in the same way as in 'Ours'. Note that the FPS axis is in the log scale.

ory update and the mask prediction steps frame-by-frame. Since the success of STM, the memory-based approach has dominated the field of semi-supervised VOS. Many variants improve STM with advanced memory read process [8, 14, 22, 34, 35] or efficient memory storage [20, 46].

One notable improvement of the memory-based approach has been made in STCN [9]. It formulates memory matching as direct image-to-image correspondence learning and proposes siamese key encoders for memory and query frames. STCN also shows that L2 similarity is more robust than inner-product for memory matching. With the advanced memory matching, STCN showcases that mem-

---

† This work was done during an internship at Adobe Research.

ory updates may not be needed at every frame. Instead, it updated the memory only at every fifth frame, resulting in considerable speedup while achieving SOTA accuracy.

Inspired by the progress, we further delve into a *per-clip inference* scheme in the memory-based approach. If we conduct the memory update periodically with an interval, we can group the input video frames into a set of consecutive frames (*i.e.* clip) according to the update interval and perform the mask prediction clip-by-clip instead of frame-by-frame. We call it *per-clip inference* (Fig. 1). This new inference scheme provides two opportunities. First, it enables us to access nearby frames before making predictions (*i.e.* non-causal), while the frame-by-frame prediction provides no access for the networks to the future frames (*i.e.* causal). With this non-causal system, we can exchange information among the frames in a clip and may make optimized predictions for the clip. To the best of our knowledge, there is no previous work in the memory-based approach that leverages clip-wise optimization. Another opportunity is the flexibility between accuracy and speed tradeoff. Increasing the memory update interval may provide near-linear speedup, since there are lower computations for memory update and, more importantly, the majority of the computations within a clip may be processed in parallel.

Based on the motivation, we present a new semi-supervised video object segmentation method, PCVOS, that is tailored for the *per-clip inference* scheme. Given the per-clip inference scenario, we propose the following changes from the standard memory-based methods. To optimize the features using intra-clip correlation, we propose *intra-clip refinement module* that performs a clip-wise operation. Specifically, we employ a transformer [39] to aggregate information in a spatial-temporal neighborhood. Since the features from the memory readout are a critical source of information for mask prediction, we place the refinement after the memory readout. The module aggregates and refines the features resulting in consistent and robust mask predictions. To enhance accuracy and speed tradeoff, we propose a *progressive memory matching mechanism*. While increasing the memory update interval provides a great opportunity for improving efficiency, we observed that memory readout accuracy is gradually degraded as the interval increases. Our progressive matching module provides a lightweight solution to augment the memory and boosts the memory readout accuracy when the memory update interval is long. In addition, we provide a new training scheme. We form each training sample with multiple *clips* and train our model with clip-level supervision. Compared to previous per-frame training [28, 29], we found that our *per-clip training* is much effective for our method.

With our new perspective and proposals, our method achieves state-of-the-art performance (*e.g.* 84.6 on Youtube-VOS 2018 val, 86.1 on DAVIS 2017 val). Fur-

thermore, by varying memory update intervals, we offer multiple variant models with great accuracy and efficiency trade-off. For example, our efficient model, *Ours-L15*, achieves better accuracy than STCN[1] while running almost three times faster as depicted in Fig. 1. More importantly, it could be possible to enjoy the flexibility with a *single* trained model via adaptive modulation of memory update interval at the test time.

Our contributions are summarized as follows:

1. We reformulate semi-supervised video object segmentation from a *per-clip inference* perspective, offering an alternative to the dominating *per-frame inference*.

2. We propose the Per-Clip VOS model (PCVOS) that is tailored for per-clip inference.

3. Our method achieves state-of-the-art performance on multiple benchmarks along with efficient variants providing great accuracy-speed balance.

2. Related work

Semi-supervised Video Object Segmentation. Early video object segmentation methods can be categorized into two groups. Online-learning methods [2, 4, 16, 23, 24, 31, 42, 45] fine-tune networks at test time to introduce target-specific information. Despite the promising results, the test-time fine-tuning is extremely time-consuming and therefore not suitable for many real-time applications.

Offline-learning methods target for learning a network that works for any videos without test-time adaptation. Under this goal, propagation-based approaches [10, 16, 23, 28, 38, 48, 49] formulate semi-supervised VOS as a temporal label propagation problem. In [28, 49], networks directly propagate object masks from the previous frame. Some methods [10, 23, 31, 37, 38, 48] utilize the optical flow for mask propagation. In general, these methods are vulnerable to occlusion and drifting, leading to error accumulation during the propagation process. More recent works [16, 23] unify the re-identification mechanism to overcome temporal discontinuities.

As another line, there are methods based on feature matching between the previous frames and the current frame [6, 15, 41, 50]. In [6, 15], a pixel-wise embedding is learned to match the current frame with the first frame with the ground-truth annotation. These methods usually are not reliable when the scene contains many similar objects and large appearance changes. FEELVOS [41] proposes to separate a global and a local matching for robustness against such challenges. In [50], background matching is additionally considered along with the attention mechanism.

Memory-based Approaches. The memory-based approach is one of the latest breakthroughs in semi-supervised

---

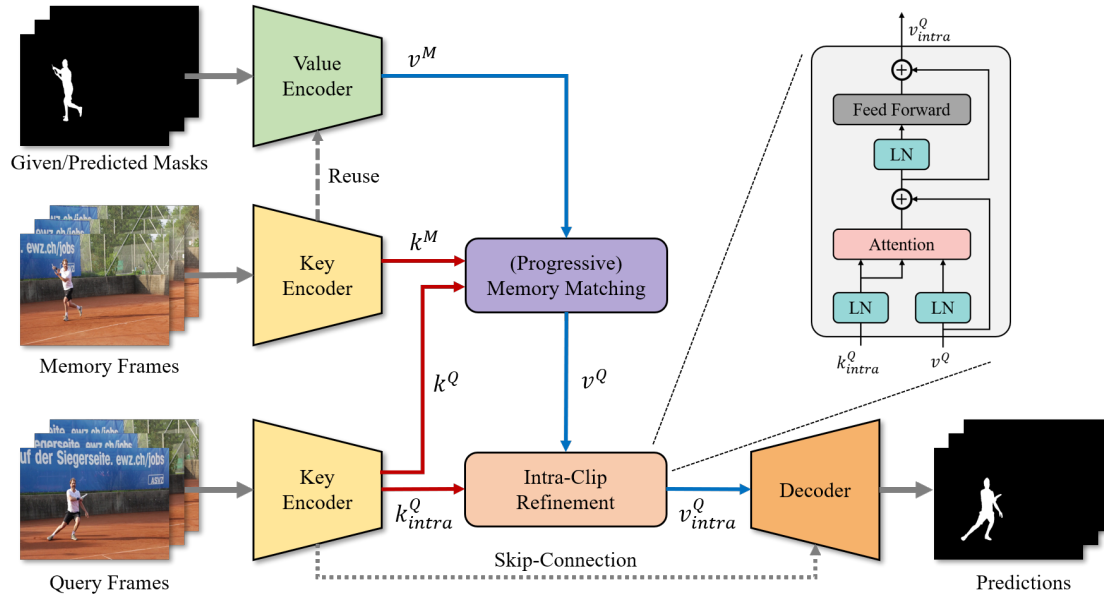[1]In Fig. 1, their original model is denoted as 'STCN-L5'.

Figure 2. **The Overview of the Proposed Framework.** The model takes multiple query frames (*i.e.* clip) as input and predicts a sequence of masks at a time. Given the memory, the memory matching module initially retrieves relevant information for all query frames and the intra-clip refinement module refines the features based on the spatio-temporal correlation among the pixels within the clip features.

video object segmentation. For the first time, STM [29] leverages a memory network to store past-frame predictions and utilize a non-local attention mechanism to read the relevant information from the memory. Many variants are proposed to improve STM in diverse aspects such as advanced memory read [8,14,22,34,35] and efficient memory storage [20,46]. KMN [34] improves memory read operation with the 2D Gaussian kernel based on memory-to-query matching. RMNet [46] only store memory of local region and conduct local-to-local matching for efficiency. LCM [14] learns object-level information and utilizes positional prior to enhance the matching accuracy. HMMN [35] proposes hierarchical memory matching that enables multi-scale memory reading. These methods process a video frame-by-frame while updating memory at every frame. However, this convention creates an upper bound of the efficiency that the memory-based approach can achieve.

Recently, STCN [9] reformulates the matching problem as pure image-based correspondence learning and shows that memory update is not needed at every frame with improved memory matching. In this work, we further study the scenario with a periodic memory update, namely *per-clip inference*. Different from STCN that processes each frame in a clip independently, we introduce a clip-wise operation that employs a transformer [39] to model spatio-temporal context *within a query clip*. Note that it is a different usage of transformer from previous VOS methods [11,26,51] that adopt a transformer mainly for improving query to memory matching (*i.e.* memory read process).

3. Proposed Framework

Given a video sequence, we divide the video into several *clips* according to the memory update interval and process each *clip* sequentially. The previous frames with predicted (or given) object mask is considered as memory and used to predict the mask of the current clip (*i.e.* query clip).

3.1. Overview

The overview of our framework is shown in Fig. 2. Our model contains five modules: 1) a key encoder that extracts key features used to build spatio-temporal correspondences between the memory and the query frames; 2) a value encoder, where the network embeds the previous mask information into value features; 3) a memory matching module that initially retrieves value information from the memory; 4) an intra-clip refinement module, where a transformer refines the retrieved value features by leveraging intra-clip correlation; 5) a decoder that takes refined information and predicts mask results. In addition, to specialize the model for the per-clip inference, we propose a new training scheme and a variant of the memory matching module, progressive memory matching mechanism. We will detail out each component in the following.

3.2. Key and Value Encoders

Overall architecture design of the key and value encoder follows STCN [9]. As depicted in Fig. 2, the previous mask information (along with memory frames) are encoded into

value features, v^M, through the value encoder. For both memory and query frames, the key encoder extracts key features, k^M and k^Q, that are used to find spatio-temporal correspondence in the memory matching module. We additionally introduce a separate branch on the key encoder to produce local key features, k^Q_{intra}, for query frames. Note that the encoders process each image (or, image and mask) independently and the features are then concatenated along the temporal dimension. Specifically, given T memory frames and L query frames, two encoders extract the following features: memory value $v^M \in \mathbb{R}^{THW \times C_v}$, memory key $k^M \in \mathbb{R}^{THW \times C_k}$, query key $k^Q \in \mathbb{R}^{LHW \times C_k}$, and query local key $k^Q_{intra} \in \mathbb{R}^{LHW \times C_{k'}}$, where HW is the spatial dimension size of the feature maps.

3.3. Memory Matching Module

As in recent VOS methods [9,14,20,29,34,46], the memory matching module first computes the pairwise similarity between all query and memory pixels in a non-local manner. Given the query key k^Q and the memory key k^M, the affinity matrix $A \in \mathbb{R}^{LHW \times THW}$ between them is computed as follows:

$$A(k^Q, k^M)_{i,j} = \frac{exp(sim(k^Q_i, k^M_j))}{\sum_j exp(sim(k^Q_i, k^M_j))}, \quad (1)$$

where sim is a similarity measure and $A_{i,j}$ denotes the affinity score at the i,j-th position. Then, each query point retrieves information in the memory value v^M based on the affinity (*i.e.* weighted sum) by:

$$v^Q = Read(k^Q, k^M, v^M) = A(k^Q, k^M)v^M. \quad (2)$$

It is worth noting that the matching process of each query point is totally independent. As the latter image within the query clip is distant from the memory in the time dimension, it is more challenging to obtain accurate correspondences due to the object deformation and motion.

3.4. Intra-Clip Refinement and Decoder

Even though the memory matching module takes the most relevant features from the memory, it is error-prone when there are new targets, occlusion, or large deformation of objects. To compensate for this, we propose to harness spatio-temporal structure across multiple query frames.

To this end, as shown in Fig. 2, we introduce the intra-clip refinement module. We adopt transformer-based attention [39] to refine the retrieved values based on the spatio-temporal correlation among the pixels in the clip. The attention layer first computes the affinity matrix among query local key k^Q_{intra}, then, the value is propagated within a clip. The retrieved value features are enhanced by the propagated values via element-wise sum. These process is summarized
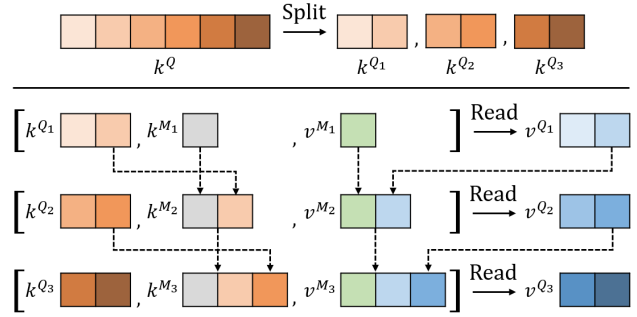

Figure 3. Proposed Progressive Memory Matching Mechanism. We illustrate an example when a clip is divided into three segments with 2 frame length (*i.e.* $S = 3$, $F = 2$).

as follows:

$$v_{attn} = A(\phi(k^Q_{intra}), \phi(k^Q_{intra}))\psi(v^Q) + v^Q \quad (3)$$

where the ϕ and ψ represent separate normalization [1] followed by linear projection layers for key and value, respectively. The feed-forward network (FFN) is kept the same as the standard. The final output of the intra-clip refinement is formulated as: $v^Q_{intra} = FFN(v_{attn}) + v_{attn}$.

As the motion of an object is continuous, the propagation across several consecutive frames could be constructed in local spatio-temporal windows. Here, we impose the locality constraint on the intra-clip refinement by adopting the 3D shifted window mechanism [21] to the attention layers. This way, we can not only largely diminish the ambiguity of correspondences but also reduce the computational cost.

Finally, the decoder takes the output of the intra-clip refinement and predicts the object mask of the query frames. Following STM [29], we gradually upsample the decoded feature and fuse it with the backbone features through a skip-connection. To handle the multi-object scenario, we use the soft-aggregation operation [28,29] to merge the predicted mask of each object.

3.5. Progressive Memory Matching Mechanism

While we are largely benefited from the spatio-temporal context in a clip, it is still difficult to find long-range correspondences as the temporal gap increases, limiting the efficiency gain we can get from the per-clip inference. To push the envelope of the tradeoff of the accuracy and efficiency gains, we propose a progressive memory matching mechanism, which is a variant of the memory matching module(Sec. 3.3). Our idea is to augment the memory temporarily using the intermediate information in the clip so that making the long-range correspondences are still accurate. To minimize the side effect, the processing should be efficient enough.

With those in mind, we split a clip into S segments with a frame interval of F and augment the memory at every F th frame. With this setting, we can still process the memory matching at each segment fully parallel. After processing each segment, we append the memory with a pair of the query key and the retrieved memory value at the last frame of the segment, as illustrated in Fig. 3. This is extremely efficient since this process bypasses all layers (*i.e.* the decoder and the value encoder) to compute the memory features and does not incur any extra computation. After processing all segments in a clip, we discard the temporary memory from the main memory.

Formally, the progressive memory matching process is summarized as follows:

$$
\begin{aligned}
v^{Q_t} = Read(k^{Q_t}, k^{M_t}, v^{M_t}) &= A(k^{Q_t}, k^{M_t})v^{M_t}, \\
\text{s.t.}\quad k^{M_t} &= Concat[k^{M_{t-1}}, last(k^{Q_{t-1}})], \\
v^{M_t} &= Concat[v^{M_{t-1}}, last(v^{Q_{t-1}})], \\
k^{M_1} = k^M, &\ v^{M_1} = v^M,
\end{aligned}
\tag{4}
$$

where v^{Q_t}, k^{Q_t} denotes the retrieved value and key of t-th query segment, and k^{M_t}, v^{M_t} represent memory key and value to produce v^{Q_t}, respectively. The final value feature v^{Q_t} is simple concatenation of each segment's output value, $v^Q = Concat[v^{Q_1}, v^{Q_2}, ..., v^{Q_S}]$.

3.6. Training Per-Clip VOS Model

Similar to previous works [9, 28–30, 34], we adopt the two-stage training: pre-training on image data and fine-tuning on video data. Our model is first trained on synthetic video samples simulated by applying random deformation on static images and corresponding object masks.

After the pre-training, we train the model on video data to learn long-range correspondences and intra-clip correlation. In our per-clip inference pipeline, we found that the previous training practices [28, 29] on video data have two main limitations to learn both functionalities: 1) Limited sample length: they sample few images (*e.g.* 3 frames) and request the model to process each frame at a time, 2) Lack of supervision signal: Only the image-level supervision signal is employed.

To solve the problem, we propose a new training pipeline tailored to our model. First, to open a possibility to learn both abilities, we pick multiple frames (*i.e.* $2N + 1$) from a video sequence. Specifically, we sample one image with the groundtruth label and two clips of length N. As shown in Fig. 2, the model sequentially processes each clip, not a frame, with the usage of previously predicted (or given) mask as the memory.

Second, we introduce clip-level supervision that aims to capture fine-grained temporal changes of objects. Specifically, given the predicted object mask $\tilde{m} \in \mathbb{R}^{KH'W'}$ and

groundtruth $m \in \mathbb{R}^{KH'W'}$, we implement the clip-level supervision with the dice coefficient [27] as follows:

$$
\mathcal{L}_{clip}(\tilde{m}, m) = \sum_{k=1}^{K} [1 - Dice(\tilde{m}^k, m^k)]
\tag{5}
$$

where $\tilde{m}^k$ denotes the predicted mask of k-th object, and K, H', W' represent the total number of objects, height, and width of the image respectively.

The final loss function is a combination of the clip-level supervision and the image level-supervision (*i.e.* cross-entropy) as: $L_{total} = L_{clip} + L_{image}$. We empirically show that the clip-level supervision allows the model to better learn both long-range correspondence and intra-clip correlation compared to the model solely relying on the image-level supervision.

4. Experiments

We experiment our model on widely used multi-object benchmarks, YoutubeVOS [47] and DAVIS 2017 [33], and single-object dataset, DAVIS 2016 [32]. To evaluate the model, we follow the standard evaluation metrics, where the region similarity $\mathcal{J}$ measures the average Intersection over Union (IoU) between the prediction and the groundtruth, and the contour accuracy $\mathcal{F}$ measures the average boundary similarity between them. We also report $\mathcal{J}$ and $\mathcal{F}$ for both seen and unseen categories on YoutubeVOS, and averaged overall score $\mathcal{J}\&\mathcal{F}$ for both datasets. We use the official evaluation servers or toolkits to obtain all the scores.

4.1. Implementation Details

For a fair comparison, we mainly follow the original details of STCN [9].

Architecture details. We instantiate the key and value encoders with ResNet50 [13] and ResNet18, respectively. We use the res4 feature, which has 1/16 resolution with respect to the input. Two ResBlocks [13] and one CBAM [44] block fuse the features of the key encoder and the value encoder to extract value features. We use the L2 similarity [9] for memory matching and set C_k and C_v to be 64 and 512. For the intra-clip refinement module, we adopt dot-product as similarity measure and use 2 layers of transformer with width 256, temporal window size 2, spatial window size 7. Thus, we set $C_{k'}$ as 256.

Training details. We leverage static image segmentation datasets [7, 17, 36, 43, 53] for pre-training. In this step, we synthesize 3 frames by applying random augmentation on a still image. Then we perform fine-tuning on video datasets, YoutubeVOS [47] and DAVIS [33]. During video training, we sample 7 frames from a video sequence (*i.e.* $N = 3$ in Sec. 3.6). The maximum temporal interval across the clip, *i.e.* inter-clip gap, is gradually increased from 5 to 15 and

annealed back to 5. To reduce the gap between training and inference, we keep the maximum temporal gap within the clip (*i.e.* intra-clip gap) to 5. The bootstrapped cross-entropy is used as the image-level supervision following [8, 9].

Inference details. We use an input size of 480p resolution for all experiments. Top-k filtering [8] is adopted for the memory matching module and with $k = 20$. We use every L-th frame as the permanent memory according to clip length L. In the progressive matching mechanism, we augment the temporary memory with the frame interval of 5 (*i.e.* $F = 5$) and these are removed when the model process the next clip. The progressive memory matching is used only during inference. We tried to include the module during training, but we observed slight performance degradation rather than improvement. We conjecture that it hinders the model to learn long-range propagation due to offering near-frame memory as a shortcut.

4.2. Ablation Study and Analysis

In this section, we provide analysis and perform extensive ablation studies on the YouTube-VOS 2019 validation set.

Component-wise Ablation. We validate the effectiveness of each component. Table 1 summarizes the results of module ablation study under different clip lengths. First, we ablate the intra-clip refinement (ICR) module to investigate the importance of communication between frames. As deteriorated results indicate, explicitly leveraging the spatio-temporal correlation is crucial for all clip length settings. We also explore the per-clip training (PCT). When we replace the training scheme with the traditional ones [9], the performance of the model is consistently degraded. It shows that the per-clip training allows the model to learn robust matching across wider temporal ranges. Lastly, we further eliminate the progressive matching mechanism (PMM). The progressive matching mechanism contributes to the performance on longer clip settings (*e.g.* $L = 15$ or 25) more than shorter settings. This implies that the progressive matching mechanism largely eases constructing long-range correspondences. Note that PMM is not used when $L = 5$. Without all the proposed methods, the model has degenerated into STCN [9]. The performance improvement ($1.9 - 4.9$ overall score) of our final model over the baselines is significant.

Effectiveness of Per-Clip Training Scheme. We investigate the impact of three factors of the proposed training scheme: 1) Type of the training scheme, 2) The number of frames used, 3) Existence of clip-level supervision.

Table 2 summarizes the results. We first change the type of training from a clip-wise to a frame-wise manner. During training, the model in Table 2-(1) predicts the masks

Method	PMM	PCT	ICR	Clip Length (L)			
				L=5	L=10	L=15	L=25
STCN [9]				82.7	81.9	79.6	78.1
	✓			82.7	82.3	81.7	81.1
	✓	✓		83.6	83.0	82.5	81.8
Ours	✓	✓	✓	**84.6**	**84.1**	**83.6**	**83.0**

Table 1. **Module ablation study under different clip length L.** PMM, PCT, and ICR denote the progressive matching mechanism, the per-clip training, and the intra-clip refinement module, respectively. For the experiments, we ablate each component sequentially.

Scheme	Type	NF	CS	Clip Length (L)			
				L=5	L=10	L=15	L=25
Trad.	Frame	3		83.1	82.6	82.3	81.8
(1)	Frame	7	✓	83.5	83.1	83.1	82.2
(2)	Clip	5	✓	83.9	83.8	83.4	82.7
	Clip	9	✓	83.8	83.3	83.3	82.7
(3)	Clip	7		83.7	83.3	82.5	82.0
Ours	Clip	7	✓	**84.6**	**84.1**	**83.6**	**83.0**

Table 2. **Ablation study on training scheme.** We vary the **Type** of training scheme (Frame-wise [28, 29] vs. Clip-wise), the number of the total frames used for training (**NF**), and adoption of clip-wise supervision (**CS**).

frame-by-frame, thus the input to the refinement module is a single image. While the performance is slightly better than one with the traditional scheme (Trad.), it is far below than Ours. This indicates that clip-wise training, where the model explicitly learns the spatio-temporal correlation, brings the performance improvements not because of simply using either multiple frames or clip-wise supervision. Next, we study the effect of the total number of frames used for training (NF). As shown in Table 2-(2), we note that all the variants show better performance over the frame-wise training, Table 2-Trad.&(1), and the best result is achieved with 7 frames. Besides, we run an experiment without clip-level supervision. As the scores indicate, we confirm that it helps to learn long-range correspondences and intra-clip correlation.

Advantages of Our Framework. Compared to the previous methods, our framework brings mainly two advantages. First, as shown in Fig. 1, our most accurate version, Ours-L5 (84.6%), largely outperforms the previous state-of-the-art method, STCN [9] (82.7%), with a similar running time. Second, we offer diverse efficient options. Our efficient variant, Ours-L15 (83.6%), still improves STCN by 0.9 of the overall score while running about three times faster (29.2 *vs* 10 FPS). It can be realized since our framework preserves the performance well even with a longer clip length. On the contrary, without all the proposed methods

Method	Overall	Seen		Unseen	
		$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
RGMP [28]	53.8	59.5	-	45.2	-
RVOS [40]	56.8	63.6	67.2	45.5	51.0
Track-Seg [5]	63.6	67.1	70.2	55.3	61.7
PReMVOS [23]	66.9	71.4	75.9	56.5	63.7
GC [18]	73.2	72.6	68.9	75.6	75.7
STM [29]	79.4	79.7	84.2	72.8	80.9
AFB-URR [20]	79.6	78.8	83.1	74.1	82.6
GraphMem [22]	80.2	80.7	85.1	74.0	80.9
GIEL [12]	80.6	80.7	85.0	75.0	81.9
CFBI [50]	81.4	81.1	85.8	75.3	83.4
KMN [34]	81.4	81.4	85.6	75.3	83.3
RMNet [46]	81.5	82.1	85.7	75.7	82.4
LWL [3]	81.5	80.4	84.9	76.4	84.4
SST [11]	81.7	81.2	-	76.0	-
CFBI+ [52]	82.0	81.2	86.0	76.2	84.6
LCM [14]	82.0	82.2	86.7	75.7	83.4
DMN-AOA [19]	82.5	82.5	86.9	76.2	82.5
HMMN [35]	82.6	82.1	87.0	76.8	84.6
STCN [9]	83.0	81.9	86.5	77.9	85.7
JOINT [25]	83.1	81.5	85.9	78.7	86.5
Ours	**84.6**	**83.0**	**88.0**	**79.6**	**87.9**

Table 3. The quantitative evaluation on the Youtube-VOS [47] 2018 validation set.

Method	Overall	Seen		Unseen	
		$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
KMN [34]	80.0	80.4	84.5	73.8	81.4
MiVOS [8]	80.3	79.3	83.7	75.3	82.8
CFBI [50]	81.0	80.6	85.1	75.2	83.0
LWL [3]	81.0	79.6	83.8	76.4	84.2
SST [11]	81.8	80.9	-	76.6	-
HMMN [35]	82.5	81.7	86.1	77.3	85.0
STCN [9]	82.7	81.1	85.4	78.2	85.9
Ours	**84.6**	**82.6**	**87.3**	**80.0**	**88.3**

Table 4. The quantitative evaluation on the Youtube-VOS [47] 2019 validation set.

Method	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
OSMN [49]	54.8	52.5	57.1
RGMP [28]	66.7	64.8	68.6
GC [18]	71.4	69.3	73.5
Track-Seg [5]	72.3	68.6	76.0
AFB-URR [20]	74.6	73.0	76.1
PReMVOS [23]	77.8	73.9	81.7
LWL [3]	81.6	79.1	84.1
STM [29]	81.8	79.2	84.3
CFBI [50]	81.9	79.1	84.6
SST [11]	82.5	79.9	85.1
GIEL [12]	82.7	80.2	85.3
GraphMem [22]	82.8	80.2	85.2
KMN [34]	82.8	80.0	85.6
CFBI+ [52]	82.9	80.1	85.7
MiVOS [8]	83.3	80.6	85.9
RMNet [46]	83.5	81.0	86.0
LCM [14]	83.5	80.5	86.5
JOINT [25]	83.5	80.8	86.2
DMN-AOA [19]	84.0	81.0	87.0
HMMN [35]	84.7	81.9	87.5
STCN [9]	85.4	82.2	88.6
Ours	**86.1**	**83.0**	**89.2**

Table 5. The quantitative evaluation on the DAVIS 2017 [33] validation set.

Method	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
OSMN [49]	73.5	74.0	72.9
RGMP [28]	81.8	81.5	82.0
PReMVOS [23]	86.8	84.9	88.6
GC [18]	86.8	87.6	85.7
RMNet [46]	88.8	88.9	88.7
STM [29]	89.3	88.7	89.9
CFBI [50]	89.4	88.3	90.5
KMN [34]	90.5	89.5	91.5
LCM [14]	90.7	**91.4**	89.9
HMMN [35]	90.8	89.6	92.0
STCN [9]	91.6	90.8	92.5
Ours	**91.9**	90.8	**93.0**

Table 6. The quantitative evaluation on the DAVIS 2016 [32] validation set.

(equivalent to STCN), the performance drastically drops as the clip length increases (see Table 1). More importantly, the variants are determined by the settings at the test time, thus a *single* model could run in multiple options and users can freely choose the options depending on the situation. For a fair comparison, we re-time the state-of-the-art methods with our hardware and report the FPS in Fig. 1.

4.3. Comparison with State-of-the-art Methods

We compare our model against state-of-the-art methods on YouTube-VOS [47], DAVIS 2017 [33], and DAVIS 2016 [32] benchmarks. Here we report the result of Ours-

$L5$ unless specified.

YouTube-VOS is the large-scale benchmark for multi-object video segmentation. It has unseen categories in the validation set, which make the YouTube-VOS benchmark good for measuring the generalization performance of algorithms. We use 474 and 507 validation videos of 2018 and 2019 versions to report the results. As shown in Table 3 and Table 4, our model significantly outperforms the state-of-the-art methods [9, 25] by **1.5** and **1.9** overall scores on the YouTube-VOS 2018 and 2019 validation sets, respectively.

DAVIS is a densely annotated video object segmentation

Figure 4. Qualitative comparisons on DAVIS 2017 validation (first video) and Youtube-VOS 2019 validation sets (second and third video). Our results show consistently better predictions compared to STCN [9] under challenging situations such as occlusion, similar objects, and appearance change.

dataset. We report our result on two versions: DAVIS 2017 and DAVIS 2016. (1) DAVIS 2017 is a multi-object extension of DAVIS 2016 and has 30 video sequences for validation. (2) DAVIS 2016 provides object-level (single-object) high-quality labels. The validation split consists of 20 videos. The experimental results on the DAVIS 2017 and DAVIS 2016 benchmarks are presented in Table 5 and Table 6, respectively. Our model achieves a mean $\mathcal{J}\&\mathcal{F}$ score of 86.1 and 91.9, which again surpass all the competitors on both DAVIS-2017 and DAVIS-2016 validation sets.

Qualitative Comparison. Fig. 4 visualizes qualitative examples of our model and STCN [9]. In the first and second videos, STCN confuses with objects and backgrounds of similar objects respectively, leading to accumulated errors. On the contrary, our model accurately discriminates the object from distractors and is robust to error drifting. In the third video, we can see that STCN fails to capture the boundary of the objects while our method produces less fragmented masks by exploiting the spatio-temporal context. All these qualitative examples confirm the proposal is effective.

5. Conclusion

In this paper, we propose a novel semi-supervised video object segmentation framework from the per-clip inference perspective. We design the framework to enjoy the two benefits of the per-clip inference: strong performance via intra-clip communication and great flexibility between speed and accuracy by modulating memory update interval. To this end, the intra-clip refinement and the progressive memory matching modules are introduced. The intra-clip refinement module aggregates information from a spatio-temporal neighborhood to refine the features. The progressive memory matching module provides an efficient solution when the memory update interval increases. In addition, to better learn both long-term correspondences and intra-clip refinement, we present the per-clip training with clip-wise supervision. Extensive experiments demonstrate that our method not only sets new state-of-the-art on multiple benchmarks but also delivers multiple efficient variants.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv:1607.06450*, 2016. 4

[2] Linchao Bao, Baoyuan Wu, and Wei Liu. Cnn in mrf: Video object segmentation via inference in a cnn-based higher-order spatio-temporal mrf. In *CVPR*, pages 5977–5986, 2018. 2

[3] Goutam Bhat, Felix Järemo Lawin, Martin Danelljan, Andreas Robinson, Michael Felsberg, Luc Van Gool, and Radu Timofte. Learning what to learn for video object segmentation. In *ECCV*, pages 777–794, 2020. 7

[4] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *CVPR*, pages 221–230, 2017. 2

[5] Xi Chen, Zuoxin Li, Ye Yuan, Gang Yu, Jianxin Shen, and Donglian Qi. State-aware tracker for real-time video object segmentation. In *CVPR*, pages 9384–9393, 2020. 7

[6] Yuhua Chen, Jordi Pont-Tuset, Alberto Montes, and Luc Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *CVPR*, pages 1189–1198, 2018. 2

[7] Ho Kei Cheng, Jihoon Chung, Yu-Wing Tai, and Chi-Keung Tang. Cascadepsp: toward class-agnostic and very high-resolution segmentation via global and local refinement. In *CVPR*, pages 8890–8899, 2020. 5

[8] 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*, pages 5559–5568, 2021. 1, 3, 6, 7

[9] 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. 1, 3, 4, 5, 6, 7, 8

[10] Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang. Segflow: Joint learning for video object segmentation and optical flow. In *ICCV*, pages 686–695, 2017. 2

[11] Brendan Duke, Abdalla Ahmed, Christian Wolf, Parham Aarabi, and Graham W Taylor. Sstvos: Sparse spatiotemporal transformers for video object segmentation. In *CVPR*, pages 5912–5921, 2021. 3, 7

[12] Wenbin Ge, Xiankai Lu, and Jianbing Shen. Video object segmentation using global and instance embedding learning. In *CVPR*, pages 16836–16845, 2021. 7

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 5

[14] Li Hu, Peng Zhang, Bang Zhang, Pan Pan, Yinghui Xu, and Rong Jin. Learning position and target consistency for memory-based video object segmentation. In *CVPR*, pages 4144–4154, 2021. 1, 3, 4, 7

[15] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G Schwing. Videomatch: Matching based video object segmentation. In *ECCV*, pages 54–70, 2018. 2

[16] Xiaoxiao Li and Chen Change Loy. Video object segmentation with joint re-identification and attention-aware mask propagation. In *ECCV*, pages 90–105, 2018. 2

[17] Xiang Li, Tianhan Wei, Yau Pun Chen, Yu-Wing Tai, and Chi-Keung Tang. Fss-1000: A 1000-class dataset for few-shot segmentation. In *CVPR*, pages 2869–2878, 2020. 5

[18] Yu Li, Zhuoran Shen, and Ying Shan. Fast video object segmentation using the global context module. In *ECCV*, pages 735–750, 2020. 7

[19] Shuxian Liang, Xu Shen, Jianqiang Huang, and Xian-Sheng Hua. Video object segmentation with dynamic memory networks and adaptive object alignment. In *ICCV*, pages 8065–8074, 2021. 7

[20] Yongqing Liang, Xin Li, Navid Jafari, and Jim Chen. Video object segmentation with adaptive feature bank and uncertain-region refinement. In *NIPS*, volume 33, 2020. 1, 3, 4, 7

[21] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *arXiv:2106.13230*, 2021. 4

[22] Xiankai Lu, Wenguan Wang, Martin Danelljan, Tianfei Zhou, Jianbing Shen, and Luc Van Gool. Video object segmentation with episodic graph memory networks. In *ECCV*, pages 661–679, 2020. 1, 3, 7

[23] Jonathon Luiten, Paul Voigtlaender, and Bastian Leibe. Premvos: Proposal-generation, refinement and merging for video object segmentation. In *ACCV*, pages 565–580, 2018. 2, 7

[24] K-K Maninis, Sergi Caelles, Yuhua Chen, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. Video object segmentation without temporal information. *IEEE TPAMI*, 41(6):1515–1530, 2018. 2

[25] Yunyao Mao, Ning Wang, Wengang Zhou, and Houqiang Li. Joint inductive and transductive learning for video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9670–9679, 2021. 7

[26] Jianbiao Mei, Mengmeng Wang, Yeneng Lin, Yi Yuan, and Yong Liu. Transvos: Video object segmentation with transformers. *arXiv:2106.00588*, 2021. 3

[27] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3DV*, pages 565–571, 2016. 5

[28] Seoung Wug Oh, Joon-Young Lee, Kalyan Sunkavalli, and Seon Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *CVPR*, pages 7376–7385, 2018. 2, 4, 5, 6, 7

[29] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, pages 9226–9235, 2019. 1, 2, 3, 4, 5, 6, 7

[30] Kwanyong Park, Sanghyun Woo, Dahun Kim, Donghyeon Cho, and In So Kweon. Preserving semantic and temporal consistency for unpaired video-to-video translation. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 1248–1257, 2019. 5

[31] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *CVPR*, pages 2663–2672, 2017. 2

[32] 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*, pages 724–732, 2016. 5, 7

[33] 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:1704.00675*, 2017. 5, 7

[34] Hongje Seong, Junhyuk Hyun, and Euntai Kim. Kernelized memory network for video object segmentation. In *ECCV*, pages 629–645, 2020. 1, 3, 4, 5, 7

[35] Hongje Seong, Seoung Wug Oh, Joon-Young Lee, Seongwon Lee, Suhyeon Lee, and Euntai Kim. Hierarchical memory matching network for video object segmentation. In *ICCV*, pages 12889–12898, 2021. 1, 3, 7

[36] Jianping Shi, Qiong Yan, Li Xu, and Jiaya Jia. Hierarchical image saliency detection on extended cssd. *PAMI*, 38(4):717–729, 2015. 5

[37] Inkyu Shin, Kwanyong Park, Sanghyun Woo, and In So Kweon. Unsupervised domain adaptation for video semantic segmentation. *arXiv preprint arXiv:2107.11052*, 2021. 2

[38] Yi-Hsuan Tsai, Ming-Hsuan Yang, and Michael J Black. Video segmentation via object flow. In *CVPR*, pages 3899–3908, 2016. 2

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017. 2, 3, 4

[40] 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*, pages 5277–5286, 2019. 7

[41] 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*, pages 9481–9490, 2019. 2

[42] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for video object segmentation. *arXiv:1706.09364*, 2017. 2

[43] Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. Learning to detect salient objects with image-level supervision. In *CVPR*, pages 136–145, 2017. 5

[44] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 5

[45] Huaxin Xiao, Jiashi Feng, Guosheng Lin, Yu Liu, and Maojun Zhang. Monet: Deep motion exploitation for video object segmentation. In *CVPR*, pages 1140–1148, 2018. 2

[46] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Shengping Zhang, and Wenxiu Sun. Efficient regional memory network for video object segmentation. In *CVPR*, pages 1286–1295, 2021. 1, 3, 4, 7

[47] 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, 5, 7

[48] Yu-Syuan Xu, Tsu-Jui Fu, Hsuan-Kung Yang, and Chun-Yi Lee. Dynamic video segmentation network. In *CVPR*, pages 6556–6565, 2018. 2

[49] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K Katsaggelos. Efficient video object segmentation via network modulation. In *CVPR*, pages 6499–6507, 2018. 2, 7

[50] Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by foreground-background integration. In *ECCV*, pages 332–348, 2020. 1, 2, 7

[51] Zongxin Yang, Yunchao Wei, and Yi Yang. Associating objects with transformers for video object segmentation. *NeurIPS*, 2021. 3

[52] Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by multi-scale foreground-background integration. *PAMI*, 2021. 1, 7

[53] Yi Zeng, Pingping Zhang, Jianming Zhang, Zhe Lin, and Huchuan Lu. Towards high-resolution salient object detection. In *ICCV*, pages 7234–7243, 2019. 5