IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 26 July 2022, accepted 18 August 2022, date of publication 24 August 2022, date of current version 31 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3201245

RESEARCH ARTICLE

Local Memory Read-and-Comparator for Video Object Segmentation

YUK HEO[1], (Student Member, IEEE), YEONG JUN KOH[2], (Member, IEEE), AND CHANG-SU KIM[1], (Senior Member, IEEE)

[1]School of Electrical Engineering, Korea University, Seoul 02841, South Korea
[2]Department of Computer Science and Engineering, Chungnam National University, Daejeon 34134, South Korea

Corresponding author: Yeong Jun Koh (yjkoh@cnu.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) funded by the Korean Government (MSIT) under Grant NRF-2021R1A4A1031864, Grant NRF-2022R1A2B5B03002310, and Grant NRF-2022R1I1A3069113.

ABSTRACT Recently, the memory-based approach, which performs non-local matching between previously segmented frames and a query frame, has led to significant improvement in video object segmentation. However, the positional proximity of the target objects between the query and the local memory (previous frame), i.e. temporal smoothness, is often neglected. There are some attempts to solve the problem, but they are vulnerable and sensitive to large movements of target objects. In this paper, we propose local memory read-and-compare operations to address the problem. First, we propose local memory read and sequential local memory read modules to explore temporal smoothness between neighboring frames. Second, we propose the memory comparator to read the global memory and local memory adaptively by comparing the affinities of the global and local memories. Experimental results demonstrate that the proposed algorithm yields more strict segmentation results than the recent state-of-the-art algorithms. For example, the proposed algorithm improves the video object segmentation performance by 0.4% and 0.5% in terms of $\mathcal{J}\&\mathcal{F}$ on the most commonly used datasets, DAVIS2016 and DAVIS2017, respectively.

INDEX TERMS Memory network, semi-supervised video object segmentation, video object segmentation.

I. INTRODUCTION

Video object segmentation (VOS) aims at cutting out objects of interest from the background in a video. It is a fundamental task to perform many computer vision techniques, including video editing and video summarization. It also takes an essential role in facilitating real-world applications such as automatic driving or augmented reality [1]. Object deformation, occlusion, and appearance change are challenging problems [2]. To overcome these issues, semi-supervised VOS, which uses a complete annotated mask at the first frame of a video to segment out the target object, has been widely researched. Recently, many semi-supervised VOS researches have been carried out with the development of deep neural networks.

Representatively, space-time memory network [3] and its following works [4], [5], [6], [7] proposed memory-based VOS algorithms and achieved outstanding performance and efficiency. By assigning a number of previously predicted frames as memory, they predict segmentation results of the query frame using the memory through the readout process, as shown in Figure 1(a). First, they construct affinities between the memory and the query frame to conduct the readout process. Affinities then transfer the encoded feature of the memory to the query frame for reliable prediction.

However, since many memory-read processes are performed in the non-local manner [8], they overlook the property of target objects that have spatiotemporal smoothness across the video. In general, there is a constraint that object movements between neighboring frames are confined. In this regard, recent studies [9], [10] attempted to deal with this continuity by performing local matching within a specific search range, but they are vulnerable to fast or large movements beyond the corresponding search range. In contrast, the proposed method adaptively readout the global and local memory to address the problem as illustrated in Figure 1(b).

The associate editor coordinating the review of this manuscript and approving it for publication was Sudipta Roy.

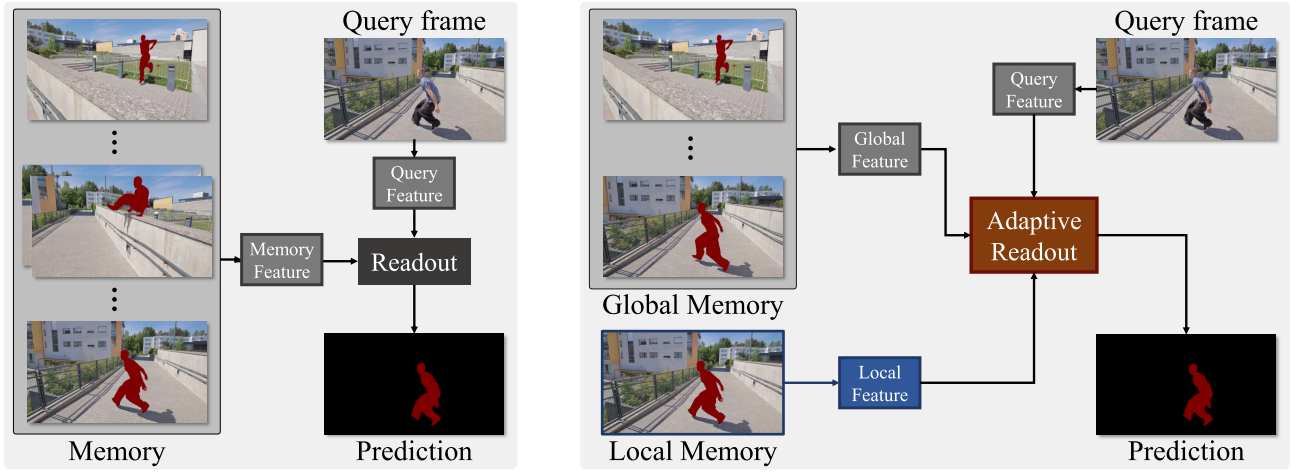

FIGURE 1. (a) Existing memory-based algorithms [4], [5], [6], [7] and (b) proposed algorithm. The proposed method adaptively reads the local information to consider the contiguity of target objects across adjacent frames.

In this paper, we propose a robust approach to achieve VOS based on the local memory read-and-comparator. First, we propose a local memory read (LMR) and sequential local memory read (SLMR) to transfer the segmentation information to neighboring frames in a hierarchical manner. Next, we design a memory comparator to read the global memory and the local memory adaptively according to the affinity between the memory frames and the query frame. Experimental results demonstrate that the proposed local memory read-and-comparator is effective and outperforms the state-of-the-arts VOS algorithms.

This paper has three main contributions:

- Effective local memory read operators to deal with spatial contiguity between adjacent frames.
- Memory comparator to selectively use the local memory and global memory.

The rest of this paper is organized as follows. Section II reviews related work on the three main approaches of video object segmentation which are unsupervised, interactive, and semi-supervised settings. Section III describes the proposed algorithm. Section IV compares the proposed algorithm with the state-of-the-art VOS algorithm and analyzes the proposed operators quantitatively and qualitatively. Finally, Section V concludes the paper.

II. RELATED WORK

A. UNSUPERVISED VOS

The objective of unsupervised VOS is to segment out primary objects in a video without any annotations or clues. Object proposals, saliency, or motion have been used before the advance of the neural networks [31], [32], [33], [34]. Recently, many deep learning-based unsupervised VOS methods [35], [36], [37], [38], [39], [40] have been introduced using the large VOS dataset [41], [42] with the improvement of parallel computing.

B. INTERACTIVE VOS

Interactive VOS aims to refine segmentation results with repeated user inputs, such as points, scribbles, or bounding boxes. A round-based interactive VOS process [43], which iterates each round of the interaction until the user is satisfied, is adopted in many recent interactive VOS algorithms [7], [44], [45], [46], [47], [48]. Cheng *et al.* [48] proposed the difference-aware fusion to fuse results of the previous round and the current round by learnable parameters. Heo *et al.* [47] introduces a guided interactive VOS system based on the reliability attention module for the annotated frame.

C. SEMI-SUPERVISED VOS

Semi-supervised VOS is a task to predict target objects in a video using an accurately and densely annotated mask at the first frame. Superpixels [49] or random walkers [50] are used for the early works. With the development of deep convolutional neural networks, VOS methods have focused on online and offline learning. Table 1 lists a summary of CNN-based semi-supervised VOS algorithms. Online learning VOS methods [11], [12], [13], [14], [15] finetune pre-trained networks with the first frame annotation of the video. Therefore, they inevitably consume additional time to train the network in inference for each video.

On the other hand, in order to eliminate the time-consuming process in online learning, offline learning algorithms have been studied based on propagation, detection, and matching. Specifically, propagation-based algorithms [16], [17], [18], [19], [20] propagate predicted masks in the previous frame to the query frame to carry out VOS. For example, AGSS [18] generated attention with the previous frame and its prediction to guide the query frame.

Matching-based algorithms [21], [22], [23], [24], [25], [26] perform the pixel-wise feature matching between query frame and other frames. For example, PML *et al.* [21]

TABLE 1. Summary table for the CNN-based semi-supervised VOS algorithms.

Algorithm name	Year	Brief Methodology	Approach	Highlights & Limitation
OSVOS [11]	2017	Proposed a fully-convolutional neural network architecture for VOS.	Online learning VOS	Finetunes the model at the inference phase to handle each object. Thus, inevitably consumes additional time to finetune VOS model in the inference phase for each video.
SegFlow [12]	2017	Designed a unified convolutional neural network that contains one branch for object segmentation and another one for optical flow.		
CTN [13]	2017	Decoded images into three branches to perform Markov random field optimization.		
OSVOS^S [14]	2018	Proposed semantic guidance to improve OSVOS by constructing information about the category of the object and the number of instances.		
CNN-MRF [15]	2018	Designed a spatiotemporal Markov random field model.		
RGMP [16]	2018	Proposed two-stage training method with static images based on Siamese encoder-decoder networks.	Propagation-based VOS	Propagates predicted masks in the previous frame to the query frame. Hard to deal with the problem when the target object disappears and reappears.
DyeNet [17]	2018	Combined template re-id and temporal propagation into a unified model.		
AGSS [18]	2019	Designed an attention-guided decoder to guide the query frame from the previous frame.		
AGAME [19]	2019	Introduced a module that generates the explicit appearance of target and background.		
TVOS [20]	2020	Diffused temporal information by utilizing long-term similarity on the target object's appearance.		
PML [21]	2018	Classified an encoded feature of each pixel based on the feature distance between the annotated frame and the query frame.	Matching-based VOS	Performs feature matching to obtain pixel-wise scores or distances between query frame and other frames. Unable to utilize features for previously segmented frames.
VideoMatch [22]	2018	Proposed a soft matching method that estimates the similarity score between different segments.		
FEELVOS [23]	2019	Trained the network in the end-to-end manner to perform the global and local matching.		
CFBI [24]	2020	Performed both pixel-level matching and instance-level matching.		
LLGC [25]	2021	Used more previously predicted frames for matching with the graph-based learning algorithm.		
CFBI+ [26]	2021	Employed multi-scale matching based on CFBI.		
STM [3]	2019	Introduced key-value memory operations on VOS task to transfer the feature from the memory to the query frame.	Memory-based VOS without temporal smoothness	Employs non-local matching to transfer target object features in the memory to the query frame. Neglected the spatiotemporal smoothness between the adjacent frame and the query frame.
GC [27]	2020	Designed global memory as fixed-size memory which is updated in every timestep.		
SST [28]	2021	Introduced transformer-based [29] VOS model.		
DMN [6]	2021	Generated object templates and aligned positional changes of target objects.		
STCN [7]	2021	Replaced the dot product to Euclidean distance of the non-local operation.		
RMNet [30]	2021	Used motion between neighboring frames to limit matching regions at the query frame.	Memory-based VOS with temporal smoothness	Addresses the temporal smoothness characteristics in video. Unable to adaptively use temporal contiguity - vulnerable to large movement or disappearance of objects in adjacent frames.
LCM [5]	2021	Encoded the relative position with sine and cosine functions.		
HMMN [9]	2021	Generated 2D Gaussian kernels which informs locality on each memory frame.		
AOT [10]	2021	Designed attention module which transfers temporally adjacent object features within the local region for each pixel based on the transformer.		

classified an encoded feature of each pixel at the query frame into foreground or background based on the feature distance between the annotated frame and the query frame. Also, [23], [24], [26] measured the pixel-wise feature distance at the query frame with the previous frame as well as the annotated frame. LLGC [25] used more unlabeled frames for matching to improve the robustness with the graph-based learning algorithm.

Recently, Oh *et al.* [3] introduced the space-time memory network (STM), which transfers the feature from the memory

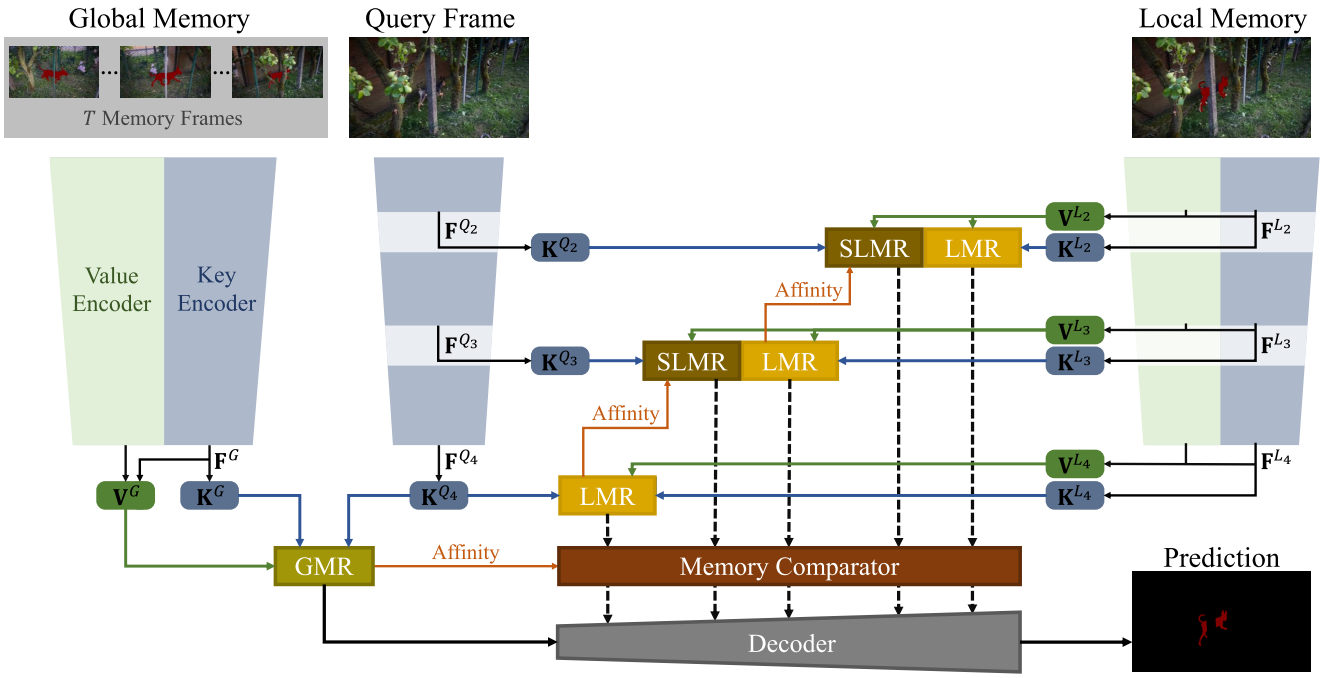

FIGURE 2. **Overview of the proposed VOS network.**

to the query frame. STM encodes several past predictions into the memory and employs non-local matching [8] to transfer target object features in the memory to the query frame. As variants of STM, many memory-based algorithms [5], [6], [7], [9], [10], [27], [28], [30] have achieved impressive performance on the semi-supervised VOS. DMN [6] generated object templates and employed a dynamic memory network to align positional changes of target objects. STCN [7] replaced the operation of matching from the dot product to Euclidean distance. In addition, some memory-based methods [5], [9], [10], [30] considered temporal smoothness between the previous frame and query frame. RMNet [30] used motion between neighboring frames to limit matching regions at the query frame. LCM [5] and AOT [10] adopted the relative positional encoding [51] with sine and cosine functions. HMMN [9] and AOT [10] transferred temporally adjacent object features by computing similarities between the query and previous frames within the local region for each pixel.

III. PROPOSED ALGORITHM

We segment out objects of interest in a video from the complete annotation at the first frame consecutively. To this end, we develop the local memory read-and-compare algorithm.

Figure 2 illustrates the overview of the proposed VOS algorithm. To predict the segmentation result at the query frame, we transfer values of the previously segmented memory frames. Given T memory frames (global memory) and the previous frame (local memory), we first apply the global memory read (GMR) operation to transfer the global memory. Then, the proposed network propagates value features of the target objects at the previous frame using the proposed LMR and SLMR operations in various resolutions. We also design the memory comparator to employ the propagated features adaptively according to the reliability of LMR and SLMR operations.

A. FEATURE EXTRACTION

1) QUERY FEATURE

We extract a key feature from the query Q using the key encoder in [7]. The key encoder takes an image as input and yields a key feature through ResNet50 [52] and a 3×3 convolution layer. Specifically, from res2'', res3'', or res4'' in ResNet50, multi-scale frame features $\mathbf{F}^{Q_r} \in \mathbb{R}^{H_r W_r \times C_r^f}$ are obtained, where $r \in \{2, 3, 4\}$ and C_r^f denote the feature stage with $1/2^r$ resolution of the input image and the number of channels at r, respectively. Then, for each feature stage r, $\mathbf{F}^{Q_r}$ is fed into the 3×3 convolution layer to obtain a key feature $\mathbf{K}^{Q_r} \in \mathbb{R}^{H_r W_r \times C_r^k}$. To this end, multi-scale query key features $\{\mathbf{K}^{Q_r}\}_{r=2}^{4}$ are extracted from the query.

2) MEMORY FEATURE

Given the global memory G and the local memory L, we extract value features as well as key features. The key features for the local memory are extracted in the same manner as the extraction of query key features. For the local memory L, multi-scale frame features $\{\mathbf{F}^{L_r}\}_{r=2}^{4}$ and key features $\{\mathbf{K}^{L_r}\}_{r=2}^{4}$ are obtained from the key encoder. Also, for value features, we encode an image and object mask jointly using

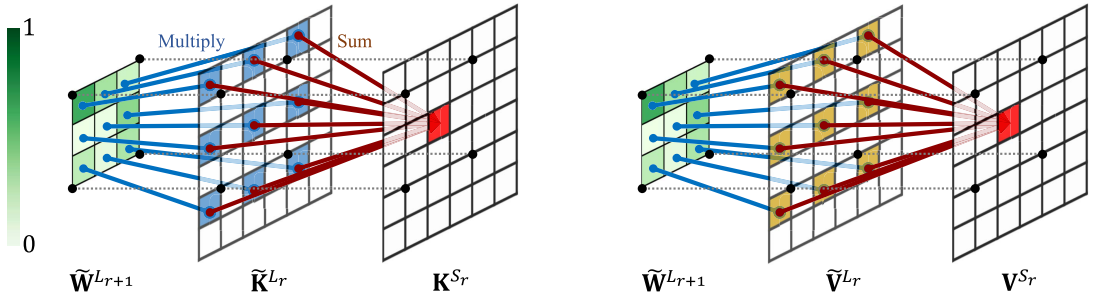

FIGURE 3. Feature reorganization in sequential local memory read (SLMR) for (a) key and (b) value at the *r*th feature stage ($d = 1$).

ResNet18, and the encoded feature is concatenated with $\mathbf{F}^{L_r}$ for each feature stage r. Finally, through a 3×3 convolution layer, a value feature $\mathbf{V}^{L_r} \in \mathbb{R}^{H_r W_r \times C_r^v}$ for the local memory is obtained for each r. On the other hand, we extract only single-scale key and value features for the global memory at the feature stage 4. Every frame in the global memory is independently embedded into key and value features, and then they are stacked along the temporal dimension to obtain a global memory key $\mathbf{K}^{G_4} \in \mathbb{R}^{TH_4W_4 \times C_4^k}$ and a global memory value $\mathbf{V}^{G_4} \in \mathbb{R}^{TH_4W_4 \times C_4^v}$ as in [7].

B. MEMORY READ OPERATOR

We employ three memory read operators, GMR, LMR, and SLMR, to predict the segmentation at the query frame from the global and local memories.

1) GLOBAL MEMORY READ (GMR)

GMR performs the equivalent role with the space-time memory read operation [3]. Given T memory frames, we obtain the value feature $\mathbf{V}^{G_4} \in \mathbb{R}^{TH_4W_4 \times C_4^v}$ and the key feature $\mathbf{K}^{G_4} \in \mathbb{R}^{TH_4W_4 \times C_4^k}$ for the global memory. GMR is designed to transfer the value feature $\mathbf{V}^{G_4}$ based on the affinity between the global key $\mathbf{K}^{G_4}$ and the query key $\mathbf{K}^{Q_4} \in \mathbb{R}^{H_4W_4 \times C_4^k}$. To this end, we first compute the global similarity matrix $\mathbf{S}^{G_4}$ by computing negative-converted L2-distance which is employed in [7] as

$$\mathbf{S}_{ij}^{G_4} = -\|\mathbf{k}_i^{Q_4} - \mathbf{k}_j^{G_4}\|_2^2, \tag{1}$$

where $\mathbf{k}_i^{Q_4}$ and $\mathbf{k}_j^{G_4}$ are feature vectors for the ith position in $\mathbf{K}^{Q_4}$ and jth position in $\mathbf{K}^{G_4}$, respectively. Then, $\mathbf{S}^{G_4}$ is normalized to obtain a global affinity matrix $\mathbf{W}^{G_4}$, which is defined as

$$\mathbf{W}_{ij}^{G_4} = \frac{\exp \mathbf{S}_{ij}^{G_4}}{\sum_{k=1}^{THW} \exp \mathbf{S}_{ik}^{G_4}}. \tag{2}$$

We compute a global readout feature $\mathbf{R}^{G_4}$ for the query via the matrix multiplication

$$\mathbf{R}^{G_4} = \mathbf{W}^{G_4} \times \mathbf{V}^{G_4}, \tag{3}$$

which can be considered as value estimation at the query frame transferred from the global memory.

2) LOCAL MEMORY READ (LMR)

We design the LMR operation to convey the segmentation information of the local memory to the query frame. Since the previous frame has more common features than any other frames to guide the query frame, especially on appearance information such as edges and boundaries, we perform LMR not only in coarse-scale key features but also in fine-scale features. For each rth feature stage, we transfer the local value feature $\mathbf{V}^{L_r} \in \mathbb{R}^{H_r W_r \times C_r^v}$ using the affinity between the local key $\mathbf{K}^{L_r} \in \mathbb{R}^{H_r W_r \times C_r^k}$ and the query key $\mathbf{K}^{Q_r}$. Unlike GMR, LMR computes the local similarity $\mathbf{S}^{L_r}$ within a local region $\mathcal{N}_i$ for each pixel i in the query to exploit spatiotemporal smoothness between neighboring frames. Specifically, $\mathbf{S}^{L_r}$ is defined as

$$\mathbf{S}_{ij}^{L_r} = \begin{cases} -\|\mathbf{k}_i^{Q_r} - \mathbf{k}_j^{L_r}\|_2^2, & j \in \mathcal{N}_i, \\ -\infty & \text{otherwise}, \end{cases} \tag{4}$$

where $\mathbf{k}_i^{Q_r}$ and $\mathbf{k}_j^{L_r}$ are feature vectors for the ith pixel in the query and jth pixel in the local memory, respectively. Also, the local region $\mathcal{N}_i$ is the set of pixels, which are sampled from $(2d+1) \times (2d+1)$ pixels around ith pixel with stride 1. The similarity is computed for those pixels in the local region only and set to infinity for the others. Then, $\mathbf{S}^{L_r}$ is normalized via the softmax operation to obtain the local affinity matrix $\mathbf{W}^{L_r}$, which has zeros values between distant pixels. Similar to GMR, a local readout feature $\mathbf{R}^{L_r}$ is obtained by

$$\mathbf{R}^{L_r} = \mathbf{W}^{L_r} \times \mathbf{V}^{L_r}. \tag{5}$$

In the LMR operation, $\mathbf{W}^{L_r}$ deals with smooth movements between adjacent frames, since it transfers the value features within $\mathcal{N}_i$ for each pixel i. Therefore, $\mathbf{R}^{L_r}$ is able to consider the space-time continuity of objects in video frames.

3) SEQUENTIAL LOCAL MEMORY READ (SLMR)

We find out that affinities between the query and local memory vary according to the level of the feature stage, even at the same position. In other words, the affinity of a pixel at

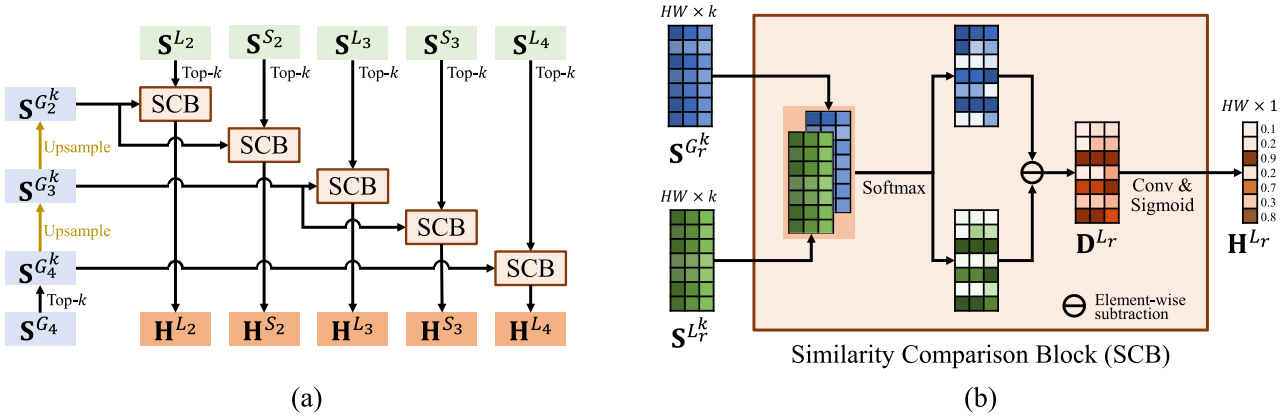

FIGURE 4. Diagrams of (a) the memory comparator and (b) the similarity comparison block (SCB).

level r is different from the affinity of the corresponding pixel at higher level $r+1$, since the key features represent different object properties according to the depth of the encoder.

Based on this observation, we propose SLMR to diversify the propagation process of the local values with higher level features. For this purpose, we reorganize the local key feature $\mathbf{K}^{L_r}$ and the local value feature $\mathbf{V}^{L_r}$ with the affinity $\mathbf{W}^{L_{r+1}}$ at the higher level (coarser scale). Let $\tilde{\mathbf{W}}^{L_{r+1}} \in \mathbb{R}^{H_{r+1}\times W_{r+1}\times H_{r+1}\times W_{r+1}}$ denote the 4D affinity tensor, which is reshaped from $\mathbf{W}^{L_{r+1}}$. Here, $\tilde{\mathbf{W}}^{L_{r+1}}(x,y,p,q)$ denotes the affinity between a pixel (x,y) in the query and a pixel (p,q) in the local memory at $r+1$th feature stage. Also, let $\tilde{\mathbf{K}}^{L_r} \in \mathbb{R}^{H_r\times W_r\times C^k}$ and $\tilde{\mathbf{V}}^{L_r} \in \mathbb{R}^{H_r\times W_r\times C^v}$ be the 3D tensors reshaped from $\mathbf{K}^{L_r}$ and $\mathbf{V}^{L_r}$, respectively. For each feature stage r, we obtain a sequential local key $\mathbf{K}^{S_r} \in \mathbb{R}^{H_r\times W_r\times C^k}$ and a sequential local value $\mathbf{V}^{S_r} \in \mathbb{R}^{H_r\times W_r\times C^v}$ using those tensors:

$$\mathbf{K}^{S_r}(x,y,c) = \sum_{u=-d}^{d}\sum_{v=-d}^{d} \tilde{\mathbf{W}}^{L_{r+1}}(\bar{x},\bar{y},\bar{x}+u,\bar{y}+v) \times \tilde{\mathbf{K}}^{L_r}(x+2u,y+2v,c) \tag{6}$$

$$\mathbf{V}^{S_r}(x,y,c) = \sum_{u=-d}^{d}\sum_{v=-d}^{d} \tilde{\mathbf{W}}^{L_{r+1}}(\bar{x},\bar{y},\bar{x}+u,\bar{y}+v) \times \tilde{\mathbf{V}}^{L_r}(x+2u,y+2v,c) \tag{7}$$

where $\bar{x} = \lceil x/2 \rceil$. This is repeated for all pixels (x,y) and channels c. As in (6) and (7), we obtain the sequential local key and value via the weighted sum with the affinity at the higher level. Figure 3 illustrates the reorganization process for $\mathbf{K}^{S_r}$ and $\mathbf{V}^{S_r}$.

$\mathbf{K}^{S_r}$ and $\mathbf{V}^{S_r}$ are reshaped to matrices. Then, the similarity matrix $\mathbf{S}^{S_r}$ and the affinity $\mathbf{W}^{S_r}$ between $\mathbf{K}^{Q_r}$ and $\mathbf{K}^{S_r}$ are sequentially computed as in LMR to acquire a sequential local readout feature

$$\mathbf{R}^{S_r} = \mathbf{W}^{S_r} \times \mathbf{V}^{S_r}. \tag{8}$$

C. MEMORY COMPARATOR

We propose the memory comparator to use readout features, which are obtained from GMR, LMR, and SLMR, adaptively. Figure 4(a) illustrates the diagram of the memory comparator. The proposed memory comparator estimates pixel-wise weights for the local readout features $\{\mathbf{R}^{L_r}\}_{r=2}^{4}$ and the sequential local readout features $\{\mathbf{R}^{S_r}\}_{r=2}^{3}$ by comparing the similarity matrix $\mathbf{S}^{G_4}$ in GMR with $\{\mathbf{S}^{L_r}\}_{r=2}^{4}$ in LMR and $\{\mathbf{S}^{S_r}\}_{r=2}^{3}$ in SLMR.

1) TOP-K SELECTION

We select top-k on each row in the similarity matrices and remove the other ones, and thus we obtain $\mathbf{S}^{G_4^k} \in \mathbb{R}^{H_4W_4\times k}$, $\{\mathbf{S}^{L_r^k} \in \mathbb{R}^{H_rW_r\times k}\}_{r=2}^{4}$, and $\{\mathbf{S}^{S_r^k} \in \mathbb{R}^{H_rW_r\times k}\}_{r=2}^{3}$. Through the top-$k$ operation, the memory comparator considers k primary similarities between the query and the memory. Since there is only one scale ($H_4 \times W_4$) for the global similarity matrix, we sequentially upsample $\mathbf{S}^{G_4^k}$ using bilinear interpolation to obtain $\{\mathbf{S}^{G_r^k} \in \mathbb{R}^{H_rW_r\times k}\}_{r=2}^{3}$.

2) SIMILARITY COMPARISON BLOCK

Similarity Comparison Block (SCB) takes a pair of the global similarity $\mathbf{S}^{G_r^k}$ and the local similarity $\mathbf{S}^{L_r^k}$ (or sequential local similarity $\mathbf{S}^{S_r^k}$) for each feature stage r. When $\mathbf{S}^{G_r^k}$ and $\mathbf{S}^{L_r^k}$ are given, SCB produces reliability weights that indicate which pixels in the local readout feature are more reliable than those in the global readout feature. When a pixel i has a larger local similarity than global similarity, SCB assigns high weight to the local readout feature for pixel i. As in Figure 4(b), SCB compares $\mathbf{S}^{L_r^k}$ and $\mathbf{S}^{G_r^k}$ via element-wise subtraction with the softmax operation. Thus, a difference map $\mathbf{D}^{L_r} \in \mathbb{R}^{H_rW_r\times k}$ is obtained by

$$\mathbf{D}_{ij}^{L_r} = \frac{\alpha}{2} \cdot \frac{\exp \mathbf{S}_{ij}^{L_r^k} - \exp \mathbf{S}_{ij}^{G_r^k}}{\exp \mathbf{S}_{ij}^{L_r^k} + \exp \mathbf{S}_{ij}^{G_r^k}} \tag{9}$$

where α is a scale factor. $\mathbf{D}^{L_r}$ is fed into a 1×1 convolution with a single output channel and the sigmoid

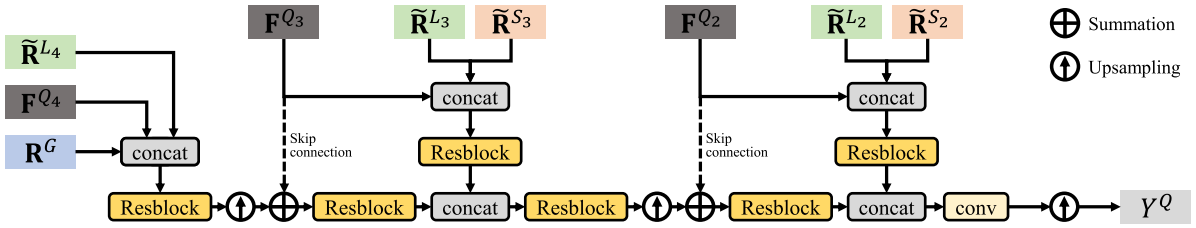

FIGURE 5. A diagram of the decoder.

operation sequentially, resulting in the reliability weight $\mathbf{H}^{L_r} \in \mathbb{R}^{H_r W_r \times 1}$. Thus, $\mathbf{H}^{L_r}$ is designed for limiting the usage of $\mathbf{R}^{L_r}$ if only $\mathbf{R}^{G_r}$ is unreliable to estimate segmentation results by comparing the similarities. Then, a weighed local readout feature $\tilde{\mathbf{R}}^{L_r}$ is given by

$$\tilde{\mathbf{R}}^{L_r} = \mathbf{H}^{L_r} \odot \mathbf{R}^{L_r} \tag{10}$$

where $\odot$ denotes that each coefficient in $\mathbf{H}^{L_r}$ is multiplied to all C_r^v coefficients in $\mathbf{R}^{L_r}$ at the same spatial positions. As in Figure 4(a), SCB is applied to both local and sequential local readout features for all feature stages. To this end, the weighed readout features $\{\tilde{\mathbf{R}}^{L_r}\}_{r=2}^{4}$ and $\{\tilde{\mathbf{R}}^{S_r}\}_{r=2}^{3}$ are obtained and fed into the decoder.

D. DECODER

Figure 5 shows the architecture of the decoder. In the decoder, features are gradually upsampled by a factor of two with the readout features, i.e. $\mathbf{R}^{G}$, $\{\tilde{\mathbf{R}}^{L_r}\}_{r=2}^{4}$, and $\{\tilde{\mathbf{R}}^{S_r}\}_{r=2}^{3}$, and frame features $\{\mathbf{F}^{Q_r}\}_{r=2}^{4}$ using skip-connections. As in Figure 5, multi-scale readout features are processed according to feature scales. Finally, the output of the final layer is upsampled by a factor of four to be of the same size as the input frame using bilinear interpolation.

E. IMPLEMENTATION DETAILS

1) LOSS

The proposed network is trained to minimize the loss

$$\mathcal{L} = \mathcal{L}_{\text{pce}} + \beta \mathcal{L}_{\text{scale}} \tag{11}$$

where $\mathcal{L}_{\text{pce}}$ is pixel-wise cross entropy in [53] between the segmentation prediction and the ground-truth. Also, the scale loss $\mathcal{L}_{\text{scale}}$ is designed to minimize query key features between different scales

$$\mathcal{L}_{\text{scale}} = \frac{1}{H_4 W_4} \sum_{i}^{H_4 W_4} \left(\|\mathbf{k}_i^{Q_4}\|_2^2 - \|\mathbf{k}_{i'}^{Q_3}\|_2^2 \right)^2 + \left(\|\mathbf{k}_i^{Q_4}\|_2^2 - \|\mathbf{k}_{i''}^{Q_2}\|_2^2 \right)^2 \tag{12}$$

where i, i', and i'' denote the equivalent position in query key features. $\mathcal{L}_{\text{scale}}$ is used until 1K iterations. We propose $\mathcal{L}_{\text{scale}}$ to boost the training of the memory comparator in the early training stage.

2) TRAINING AND INFERENCE

For training, we use training videos in DAVIS2017 [41] and YouTube-VOS [42] to train the proposed model. We randomly select three different frames within 10 frames: one for the global memory, another for the local memory, and the other for the query frame. We set the mini-batch size to 8. We use the Adam optimizer [54]. The training is repeated 200K iterations with an RTX 3090 GPU. We initialize the key encoder and the value encoder with the pre-trained weights in STCN [7]. In inference, every 5th frame except the previous frame is picked for the global memory, and the previous frame is used for the local memory.

3) PARAMETERS

The channel dimensions C_2^f, C_3^f, and C_4^f are set to 256, 512, and 1024, respectively. The dimension of key features C_2^k, C_3^k, and C_4^k are equally set to 64. For value features, the number of channels C_2^v C_3^v, and C_4^v are set to 64, 128, and 256, respectively. Also, we experimentally decide the offset of the local region $d = 2$, top-5 in the memory comparator, $\alpha = 3$ in (9), and $\beta = 10^{-4}$ in (11).

4) MEMORY MANAGEMENT IN LMR AND SLMR

Since LMR and SLMR are performed in fine scales as well as coarse scales, constructing similarities and affinities for each feature stage may lead to memory overflow. In order to prevent this issue, we construct the local similarities and affinities to store valid values. Since the number of the validate values for the similarities and affinities in each pixel is $(2d+1)^2$, memory complexity requires only $\mathcal{O}(H_r W_r \cdot (2d+1)^2)$ instead of $\mathcal{O}(H_r W_r \cdot H_r W_r)$ at feature stage r.

IV. EXPERIMENTAL RESULTS

In this section, we first compare the proposed algorithm with the state-of-the-art VOS algorithms on various datasets. Second, we analyze the proposed local read operations and memory comparator through various ablation studies.

A. DATASETS

1) DAVIS

DAVIS [2], [41] is a densely annotated VOS dataset, which is the most commonly used to evaluate VOS algorithms. It provides 480p videos in two separate datasets: DAVIS2016 and DAVIS2017. DAVIS2016 provides single-object annotated 50 videos, which are divided into 30 for training and 20 for validation. DAVIS2017 provides 60/30/30 videos

TABLE 2. Comparison of the proposed algorithm with the state-of-the-art VOS algorithms on the DAVIS2016 and DAVIS2017 validation sets. The best results are boldfaced. †: selection of ResNet50 backbone for the fair comparison.

	DAVIS2016			DAVIS2017		
	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
STM[ICCV19] [3]	89.3	88.7	89.9	81.9	79.3	84.5
LWL[ECCV20] [55]	-	-	-	81.6	79.1	84.1
CFBI[ECCV20] [24]	89.4	88.3	90.5	81.9	79.1	84.6
EGMN[ECCV20] [56]	-	-	-	82.8	80.2	85.2
KMN[ECCV20] [4]	90.5	89.5	91.5	82.8	80.0	85.6
CFBI+[TPAMI21] [26]	89.9	88.7	91.1	82.9	80.1	85.7
SSTVOS[CVPR21] [28]	-	-	-	82.5	79.9	85.1
RMNet[CVPR21] [30]	88.8	88.9	88.7	83.5	81.0	86.0
LCM[CVPR21] [5]	90.7	89.9	91.4	83.5	80.5	86.5
MiVOS[CVPR21] [48]	91.0	89.9	92.2	83.3	80.6	85.9
JOINT[ICCV21] [57]	-	-	-	83.5	80.8	86.2
DMN†[ICCV21] [6]	-	-	-	84.0	81.0	87.0
HMMN[ICCV21] [9]	90.8	89.6	92.0	84.7	81.9	87.5
AOT†[NIPS21] [10]	91.1	90.1	92.1	84.9	82.3	87.5
STCN[NIPS21] [7]	91.7	90.4	93.0	85.3	82.0	88.6
LMRC (Proposed)	**92.1**	**90.8**	**93.3**	**85.8**	**82.6**	**89.1**

TABLE 3. Comparison of the proposed algorithm with the state-of-the-art VOS algorithms on the YouTube2018 and YouTube2019 validation sets. †: selection of ResNet50 backbone for the fair comparison.

	YouTube2018					YouTube2019				
	$\mathcal{G}$	$\mathcal{J}_S$	$\mathcal{F}_S$	$\mathcal{J}_U$	$\mathcal{F}_U$	$\mathcal{G}$	$\mathcal{J}_S$	$\mathcal{F}_S$	$\mathcal{J}_U$	$\mathcal{F}_U$
STM [3]	79.4	79.7	84.2	72.8	80.9	79.2	79.6	83.6	73.0	80.6
LWL [55]	81.5	80.4	84.9	76.4	84.4	81.0	79.6	83.8	76.4	84.2
CFBI [24]	81.4	81.1	85..8	75.3	83.4	81.0	80.6	85.1	75.2	83.0
EGMN [56]	80.2	80.7	85.1	74.0	80.9	-	-	-	-	-
KMN [4]	81.4	81.4	85.6	75.3	83.3	80.0	80.4	73.8	84.5	81.4
CFBI+ [26]	82.0	81.2	86.0	76.2	84.6	82.6	81.7	86.2	77.1	85.2
SSTVOS [28]	81.7	81.2	-	76.0	-	81.8	80.9	-	76.6	-
MiVOS [48]	80.4	80.0	84.6	74.8	82.4	80.4	80.0	84.6	74.8	82.4
LCM [5]	82.0	82.2	86.7	75.7	83.4	-	-	-	-	-
RMNet [30]	81.5	82.1	85.7	75.7	82.4	-	-	-	-	-
JOINT [57]	83.1	81.5	85.9	78.7	86.5	82.8	80.8	84.8	79.0	86.6
DMN† [6]	82.5	82.5	86.9	76.2	84.2	-	-	-	-	-
HMMN [9]	82.6	82.1	87.0	76.8	84.6	82.5	81.7	86.1	77.3	85.0
AOT† [10]	84.1	**83.7**	**88.5**	78.1	86.1	84.1	**83.5**	**88.1**	78.4	86.3
STCN [7]	84.3	83.2	87.9	79.0	87.3	84.2	82.6	87.0	79.4	87.7
LMRC (Proposed)	**84.5**	83.2	87.9	**79.2**	**87.6**	**84.2**	82.5	86.9	**79.6**	**87.9**

for training/validation/test-dev sets with multi-object annotations. Region similarity $\mathcal{J}$, contour accuracy $\mathcal{F}$, and their mean $\mathcal{J}\&\mathcal{F}$ are used as metrics in experiments.

2) YouTube-VOS

YouTube-VOS [42] is the large-scale VOS dataset. It provides 3471 training videos and 474/507 validation videos for YouTube2018/YouTube2019 datasets with multi-object annotations in various resolutions. In our evaluation, we resize the input frames to have a resolution of 480p. It has 65 seen and 26 unseen object categories. We measure $\mathcal{J}_S$ and $\mathcal{F}_S$ for the seen categories and $\mathcal{J}_U$ and $\mathcal{F}_U$ for the unseen categories. We also use the overall score $\mathcal{G}$, which is the mean of the four metrics.

B. COMPARATIVE ASSESSMENT

1) DAVIS

Table 2 compares the proposed algorithm with the existing semi-supervised VOS algorithms on the validation sets in DAVIS2016 and DAVIS2017. Scores in Table 2 are from the respective papers. For DAVIS2016, the proposed algorithm improves the segmentation performance by 0.4%, 0.4%, and 0.3% in terms of $\mathcal{J}\&\mathcal{F}$, $\mathcal{J}$, and $\mathcal{F}$, respectively. Also, For DAVIS2017, in spite of its difficulty, the proposed algorithm achieves performance improvements of 0.5%, 0.6%, and 0.5% in terms of $\mathcal{J}\&\mathcal{F}$, $\mathcal{J}$, and $\mathcal{F}$. This indicates that the proposed local read-and-comparator model is effective for both single object and multiple object cases.

2) YouTube-VOS

Table 3 shows the comparison of the proposed algorithm with the existing VOS algorithms on the YouTube2018 and YouTube2019 validation sets. In terms of $\mathcal{G}$, the proposed algorithm achieves the best performance on YouTube2018 and the same performance as the state-of-the-art [7] on YouTube2019. Specifically, for the seen categories, the proposed algorithm stands second and third place on

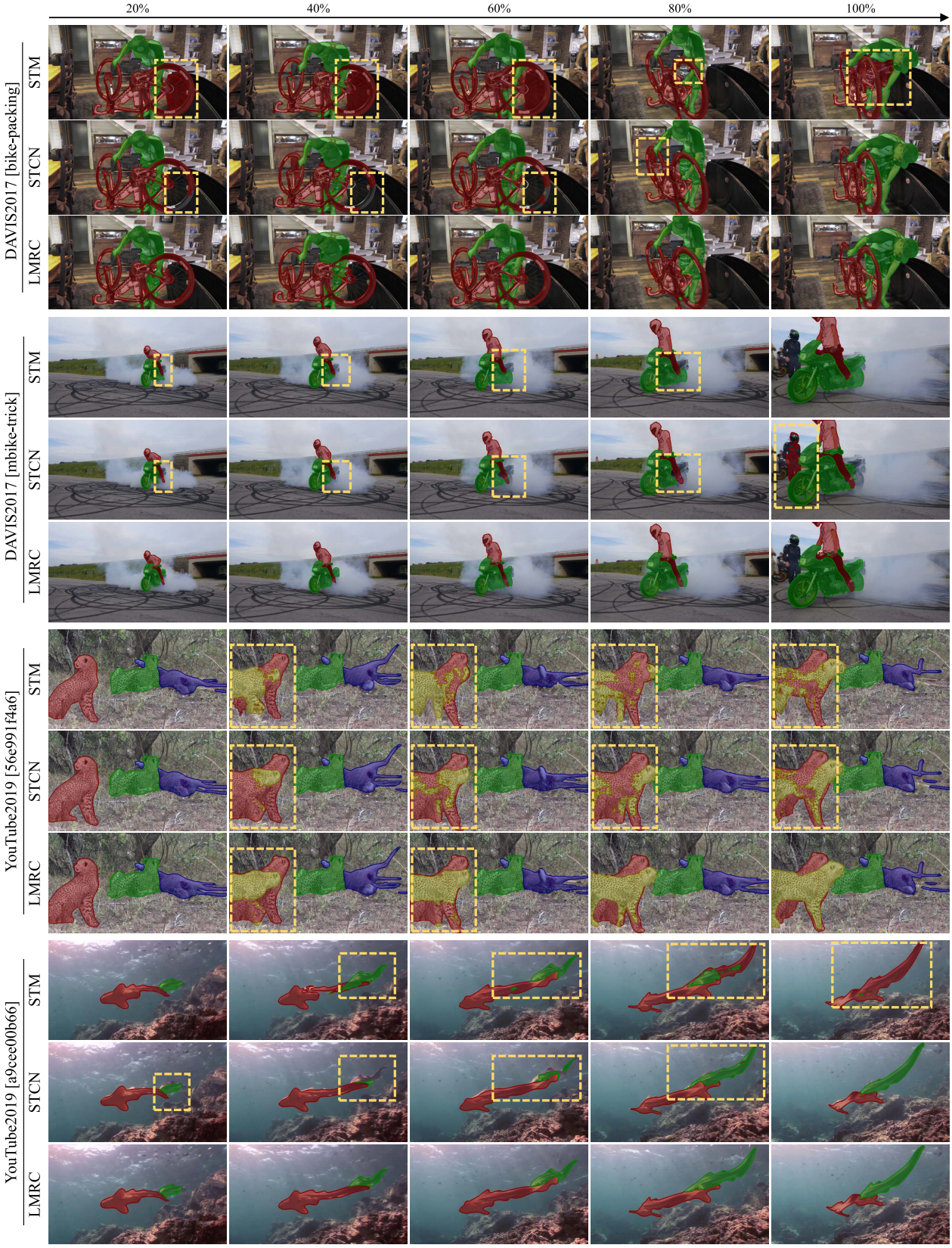

FIGURE 6. Qualitative comparison on DAVIS2017 and YouTube2019 validation sets. We compare the proposed algorithm (LMRC) with STM [3] and STCN [7]. Failed predictions are marked in yellow boxes with the dotted line.

TABLE 4. Ablation study results on DAVIS2017 and YouTube2018 validation sets. The best results are boldfaced, and the second-best ones are underlined.

Case	LMR-S	LMR	SLMR	MC	DAVIS2017 $\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	fps	YouTube2018 $\mathcal{G}$	$\mathcal{J}_S$	$\mathcal{F}_S$	$\mathcal{J}_U$	$\mathcal{F}_U$
A					83.0	79.8	86.3	15.5	82.7	82.7	87.4	76.0	84.6
B	✓				84.6	81.1	88.1	11.4	83.8	82.5	87.1	78.8	87.1
C		✓			84.9	81.6	88.1	9.4	84.1	82.8	87.4	79.0	87.3
D		✓	✓		85.1	81.9	88.3	7.9	83.9	**83.5**	**88.1**	77.9	86.0
E		✓		✓	<u>85.4</u>	<u>82.2</u>	<u>88.7</u>	9.2	<u>84.2</u>	82.9	87.7	<u>79.0</u>	<u>87.5</u>
F		✓	✓	✓	**85.8**	**82.6**	**89.1**	7.7	**84.5**	<u>83.2</u>	<u>87.9</u>	**79.2**	**87.6**

TABLE 5. Ablation studies of the hyper-parameters.

d ($\mathcal{N}$)	YT18 $\mathcal{G}$	DAVIS17 $\mathcal{J}\&\mathcal{F}$	fps
1 (3×3)	84.3	**85.8**	8.4
2 (5×5)	**84.5**	**85.8**	7.7
3 (7×7)	84.4	**85.8**	6.9
4 (9×9)	84.3	**85.8**	6.1

(a) d for the local region $\mathcal{N}$

k	YT18 $\mathcal{G}$	DAVIS17 $\mathcal{J}\&\mathcal{F}$
1	83.2	85.3
3	83.6	85.5
5	**84.5**	**85.8**
7	84.1	85.3

(b) top-k selection

YouTube2018 and YouTube2019, respectively. On the other hand, we observe that the proposed method shows the best segmentation results for the unseen categories on both YouTube2018 and YouTube2019. This indicates that the proposed method has superior generalization performance as compared with the state-of-the-arts. The proposed local read operations and memory comparator are robust to unseen categories by exploiting spatiotemporal smoothness between neighboring frames.

3) QUALITATIVE COMPARISON

Figure 6 shows qualitative comparison with STM [3] and STCN [7] on DAVIS2017 and YouTube2019 validation sets. Both STM and STCN fail to accurately segment out detailed regions such as bike wheels on 'bike-packing' and 'mbike-trick' sequences. Also, they are vulnerable to overlapped objects of the same category as in the YouTube-VOS examples. In '56e991f4a6' sequence, they failed to recognize the boundaries of the two overlapping cheetahs. In 'a9cee00b66' sequence, STM even merged them into one object in the end. On the other hand, the proposed algorithm (LMRC) provides accurate results by exploiting the local memory effectively.

C. ANALYSIS

1) ABLATION STUDY

We first analyze the effectiveness of the proposed components: LMR, SLMR, and memory comparator (MC). In table 4, we report $\mathcal{J}\&\mathcal{F}$, $\mathcal{J}$, $\mathcal{F}$scores, and frame per second (fps) for various settings on the DAVIS2017 validation set. We also measure $\mathcal{G}$, $\mathcal{J}_S$, $\mathcal{F}_S$, $\mathcal{J}_U$, and $\mathcal{F}_U$ on the YouTube2018 validation set. We trained each case in the same manner in III-E.

Setting A is the baseline, which uses GMR only. In setting B, LMR is employed for only a single scale at 4th feature stage, which is denoted as LMR-S. Settings B and C show that LMR improves performance. Also, the performance gap between B and C indicates that multi-scale readout features are effective in transferring the information of the local memory to the query. In addition, we see that LMR dramatically increases the performance of the unseen categories on YouTube2018. It is because LMR effectively transfers features within the local region and the local readout feature is trained to emphasize more on the pixel-level than category-level. We also observe that SLMR effectively increases the accuracy of segmentation results from setting D and F. Note that SLMR lowers the overall performance without the memory comparator, but improves the performance for both seen and unseen categories with the memory comparator on YouTube2018. Finally, settings E and F outperform setting C and D, respectively, by employing the proposed memory comparator commonly. Thus, these results demonstrate that the memory comparator significantly improves the performance, which requires little time.

2) LOCAL REGION AND TOP-K SELECTION

We analyze the local region of LMR and SLMR, and the top-k selection in the memory comparator on YouTube2018 and DAVIS2017 validation sets. Table 5(a) shows that $d = 2$ provides the best performance. Also, we observe that there are no significant changes according to the size of the local region. This is because LMR and SLMR are adaptively used based on the reliability weights. Table 5(b) shows how the performance is varying as k changes. $k = 5$ yields the best performance on both datasets.

3) RELIABILITY WEIGHT

Figure 7 shows the reliability weights $\mathbf{H}^{L_3}$ and $\mathbf{H}^{L_4}$, provided by the memory comparator, for three scene cases: static, dynamic, and fast movement. We observe three properties of the reliability weight. First, $\mathbf{H}^{L_3}$ has high-reliability weights

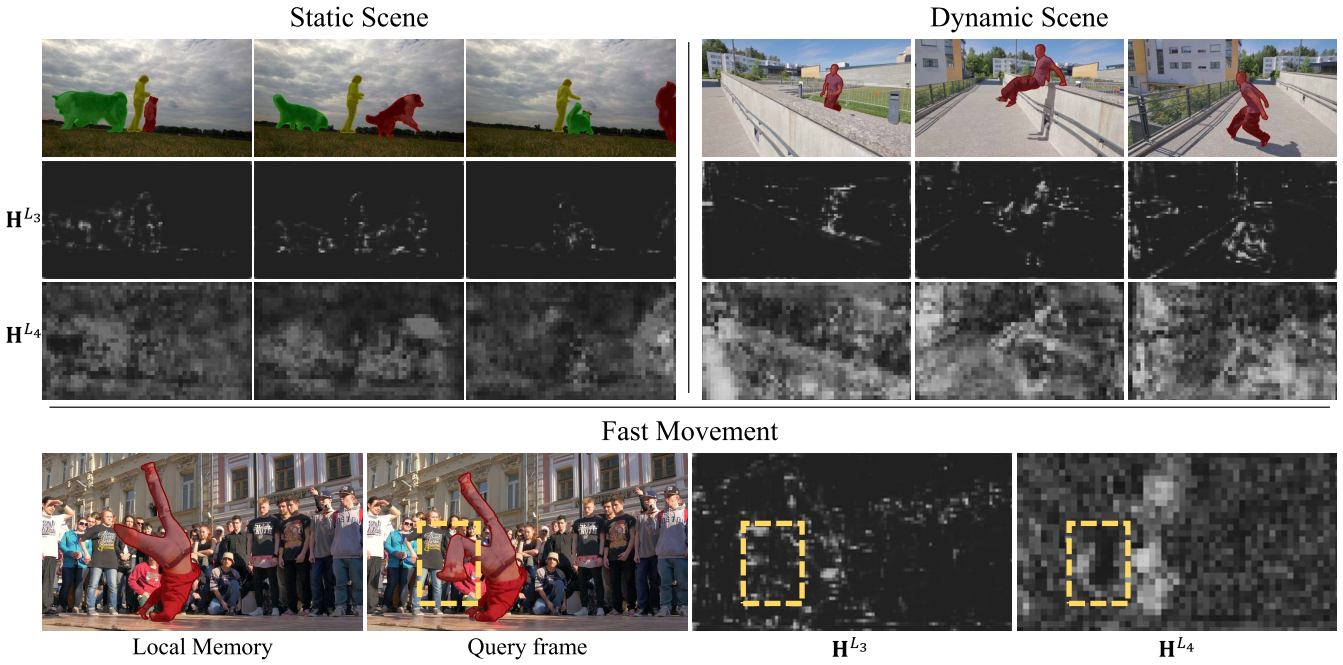

FIGURE 7. Visualization of the reliability weights $\mathbf{H}^{L_3}$ and $\mathbf{H}^{L_4}$ for three scene cases.

near object edges, which indicates that local readout features are intensely used on object edges to deal with spatiotemporal smoothness motions of target objects between adjacent frames. Second, $\mathbf{H}^{L_4}$ maps in the dynamic scene are generally higher than the static scene. In a static scene, the global readout features are sufficiently reliable since frames in the global memory have similar features to each other. On the other hand, the global readout features in dynamic scenes are generally unreliable, and thus the local readout features should be used with high weights. From $\mathbf{H}^{L_4}$ maps in dynamic and static scenes, we can observe that the proposed memory comparator provides effective reliability maps for accurate segmentation. Third, the memory comparator effectively filters out the local readout features at fast-moving regions of the object (right leg within the yellow box) with low-reliability weight. Thus, the memory comparator deals with the problem of large movements out of the local region $\mathcal{N}$.

V. CONCLUSION

We proposed a novel VOS algorithm that propagates the fused readout features of the local and global memories. First, we developed LMR and SLMR to convey the segmentation data hierarchically to deal with spatial proximity between adjacent frames. Second, we designed the memory comparator to adaptively read the local memory by comparing similarities of the local memory and the global memory. Experimental results demonstrated that the proposed algorithm outperforms the recent state-of-the-art algorithms and overcomes the limitation of the existing memory-based approaches. Although the proposed method is capable of using the adjacent frames, the frames of two or more frames behind should also be taken into account together as local frames with global memory, discriminatively. In the future, we will design to fuse the multiple local frames with global memory to deal with spatial contiguity.

REFERENCES

[1] R. Yao, G. Lin, S. Xia, J. Zhao, and Y. Zhou, "Video object segmentation and tracking: A survey," *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 4, pp. 1–47, 2020.
[2] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 724–732.
[3] S. W. Oh, J.-Y. Lee, N. Xu, and S. J. Kim, "Video object segmentation using space-time memory networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9226–9235.
[4] H. Seong, J. Hyun, and E. Kim, "Kernelized memory network for video object segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 629–645.
[5] L. Hu, P. Zhang, B. Zhang, P. Pan, Y. Xu, and R. Jin, "Learning position and target consistency for memory-based video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4144–4154.
[6] S. Liang, X. Shen, J. Huang, and X.-S. Hua, "Video object segmentation with dynamic memory networks and adaptive object alignment," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 8065–8074.
[7] H. K. Cheng, Y.-W. Tai, and C.-K. Tang, "Rethinking space-time networks with improved memory coverage for efficient video object segmentation," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2021, pp. 1–14.
[8] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 7794–7803.
[9] H. Seong, S. W. Oh, J.-Y. Lee, S. Lee, S. Lee, and E. Kim, "Hierarchical memory matching network for video object segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 12889–12898.
[10] Z. Yang, Y. Wei, and Y. Yang, "Associating objects with transformers for video object segmentation," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2021, pp. 2491–2502.

[11] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixe, D. Cremers, and L. Van Gool, "One-shot video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 221–230.

[12] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang, "SegFlow: Joint learning for video object segmentation and optical flow," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 686–695.

[13] W.-D. Jang and C.-S. Kim, "Online video object segmentation via convolutional trident network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5849–5858.

[14] K.-K. Maninis, S. Caelles, Y. Chen, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool, "Video object segmentation without temporal information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 6, pp. 1515–1530, Jun. 2019.

[15] L. Bao, B. Wu, and W. Liu, "CNN in MRF: Video object segmentation via inference in a CNN-based higher-order spatio-temporal MRF," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5977–5986.

[16] S. W. Oh, J.-Y. Lee, K. Sunkavalli, and S. J. Kim, "Fast video object segmentation by reference-guided mask propagation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7376–7385.

[17] X. Li and C. C. Loy, "Video object segmentation with joint re-identification and attention-aware mask propagation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 90–105.

[18] H. Lin, X. Qi, and J. Jia, "AGSS-VOS: Attention guided single-shot video object segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3949–3957.

[19] J. Johnander, M. Danelljan, E. Brissman, F. S. Khan, and M. Felsberg, "A generative appearance model for end-to-end video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8953–8962.

[20] Y. Zhang, Z. Wu, H. Peng, and S. Lin, "A transductive approach for video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6949–6958.

[21] Y. Chen, J. Pont-Tuset, A. Montes, and L. V. Gool, "Blazingly fast video object segmentation with pixel-wise metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1189–1198.

[22] Y.-T. Hu, J.-B. Huang, and A. G. Schwing, "VideoMatch: Matching based video object segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 54–70.

[23] P. Voigtlaender, Y. Chai, F. Schroff, H. Adam, B. Leibe, and L.-C. Chen, "FEELVOS: Fast end-to-end embedding learning for video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9481–9490.

[24] Z. Yang, Y. Wei, and Y. Yang, "Collaborative video object segmentation by foreground-background integration," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 332–348.

[25] H. Liang, L. Liu, Y. Bo, and C. Zuo, "Semi-supervised video object segmentation based on local and global consistency learning," *IEEE Access*, vol. 9, pp. 127293–127304, 2021.

[26] Z. Yang, Y. Wei, and Y. Yang, "Collaborative video object segmentation by multi-scale foreground-background integration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 4701–4712, Sep. 2021.

[27] Y. Li, Z. Shen, and Y. Shan, "Fast video object segmentation using the global context module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 735–750.

[28] B. Duke, A. Ahmed, C. Wolf, P. Aarabi, and G. W. Taylor, "SSTVOS: Sparse spatiotemporal transformers for video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5912–5921.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 1–11.

[30] H. Xie, H. Yao, S. Zhou, S. Zhang, and W. Sun, "Efficient regional memory network for video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1286–1295.

[31] W. Wang, J. Shen, and F. Porikli, "Saliency-aware geodesic video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3395–3402.

[32] A. Papazoglou and V. Ferrari, "Fast object segmentation in unconstrained video," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1777–1784.

[33] Y. J. Koh and C.-S. Kim, "Primary object segmentation in videos based on region augmentation and reduction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7417–7425.

[34] Y. J. Koh, Y.-Y. Lee, and C.-S. Kim, "Sequential clique optimization for video object segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 517–533.

[35] S. D. Jain, B. Xiong, and K. Grauman, "FusionSeg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2126.

[36] Z. Yang, Q. Wang, L. Bertinetto, S. Bai, W. Hu, and P. Torr, "Anchor diffusion for unsupervised video object segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 931–940.

[37] W. Wang, H. Song, S. Zhao, J. Shen, S. Zhao, S. C. H. Hoi, and H. Ling, "Learning unsupervised video object segmentation through visual attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3064–3074.

[38] A. Athar, S. Mahadevan, A. Osep, L. Leal-Taixé, and B. Leibe, "STEm-Seg: Spatio-temporal embeddings for instance segmentation in videos," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 158–177.

[39] M. Zhen, S. Li, L. Zhou, J. Shang, H. Feng, T. Fang, and L. Quan, "Learning discriminative feature with CRF for unsupervised video object segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 445–462.

[40] T. Zhou, J. Li, X. Li, and L. Shao, "Target-aware object discovery and association for unsupervised video multi-object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6985–6994.

[41] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 2017, pp. 1–6.

[42] 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 *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 603–619.

[43] S. Caelles, A. Montes, K.-K. Maninis, Y. Chen, L. Van Gool, F. Perazzi, and J. Pont-Tuset, "The 2018 DAVIS challenge on video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 2018, pp. 1–4.

[44] S. W. Oh, J.-Y. Lee, N. Xu, and S. J. Kim, "Space-time memory networks for video object segmentation with user guidance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 442–455, Jan. 2022.

[45] Y. Heo, Y. J. Koh, and C.-S. Kim, "Interactive video object segmentation using global and local transfer modules," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 297–313.

[46] J. Miao, Y. Wei, and Y. Yang, "Memory aggregation networks for efficient interactive video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10366–10375.

[47] Y. Heo, Y. J. Koh, and C.-S. Kim, "Guided interactive video object segmentation using reliability-based attention maps," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7322–7330.

[48] H. K. Cheng, Y.-W. Tai, and C.-K. Tang, "Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5559–5568.

[49] S. D. Jain and K. Grauman, "Supervoxel-consistent foreground propagation in video," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 656–671.

[50] W.-D. Jang and C.-S. Kim, "Semi-supervised video object segmentation using multiple random Walkers," in *Proc. Brit. Mach. Vis. Conf (BMVC)*, 2016, pp. 1–13.

[51] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 213–229.

[52] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[53] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1395–1403.

[54] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–15.

[55] G. Bhat, F. J. Lawin, M. Danelljan, A. Robinson, M. Felsberg, L. V. Gool, and R. Timofte, "Learning what to learn for video object segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 777–794.

[56] X. Lu, W. Wang, M. Danelljan, T. Zhou, J. Shen, and L. V. Gool, "Video object segmentation with episodic graph memory networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 661–679.
[57] Y. Mao, N. Wang, W. Zhou, and H. Li, "Joint inductive and transductive learning for video object segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9670–9679.

YUK HEO (Student Member, IEEE) received the B.S. degree in electrical engineering from Korea University, Seoul, South Korea, in 2018, where he is currently pursuing the Ph.D. degree in electrical engineering. His research interests include computer vision and machine learning and especially in the problems of video object segmentation.

YEONG JUN KOH (Member, IEEE) received the B.S. and Ph.D. degrees in electrical engineering from Korea University, Seoul, South Korea, in 2011 and 2018, respectively. In March 2019, he joined as an Assistant Professor with the Department of Computer Science and Engineering, Chungnam National University, where he is currently a Professor. His research interests include computer vision and machine learning, especially in the problems of video object discovery and segmentation.

CHANG-SU KIM (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from Seoul National University, in 2000. From 2000 to 2001, he was a Visiting Scholar with the Signal and Image Processing Institute, University of Southern California, Los Angeles. From 2001 to 2003, he coordinated the National Research Laboratory for 3D Visual Information Processing, 3D Data Compression Group, SNU. From 2003 to 2005, he was an Assistant Professor with the Department of Information Engineering, The Chinese University of Hong Kong. In September 2005, he joined the School of Electrical Engineering, Korea University, where he is also a Professor. He has published more than 300 journals and conference papers. His research interests include image processing, computer vision, and machine learning. In 2009, he received the IEEK/IEEE Joint Award for Young IT Engineer of the Year. In 2014, he received the Best Paper Award from *Journal of Visual Communication and Image Representation* (JVCI). He is a member of the Multimedia Systems & Application Technical Committee (MSATC) of the IEEE Circuits and Systems Society. Also, he was an APSIPA Distinguished Lecturer for term (2017–2018). He served as an Editorial Board Member of *JVCI*, an Associate Editor of IEEE TRANSACTIONS ON IMAGE PROCESSING, and an Associate Editor of IEEE TRANSACTIONS ON MULTIMEDIA. He is a Senior Area Editor of *JVCI*. He received the Distinguished Dissertation Award for Ph.D. degree, in 2000.