Guided Co-Segmentation Network for Fast Video Object Segmentation

Weide Liu, Guosheng Lin, Tianyi Zhang, and Zichuan Liu

***Abstract*—Semi-supervised video object segmentation is a task of propagating instance masks given in the first frame to the entire video. It is a challenging task since it usually suffers from heavy occlusions, large deformation, and large variations of objects. To alleviate these problems, many existing works apply time-consuming techniques such as fine-tuning, post-processing, or extracting optical flow, which makes them intractable for online segmentation. In our work, we focus on online semi-supervised video object segmentation. We propose a GCSeg (Guided Co-Segmentation) Network which is mainly composed of a Reference Module and a Co-segmentation Module, to simultaneously incorporate the short-term, middle-term, and long-term temporal inter-frame relationships. Moreover, we propose an Adaptive Search Strategy to reduce the risk of propagating inaccurate segmentation results in subsequent frames. Our GCSeg network achieves state-of-the-art performance on online semi-supervised video object segmentation on Davis 2016 and Davis 2017 datasets.**

***Index Terms*—Video Segmentation, Co-Segmentation, Semi-supervised**

I. Introduction

Video object segmentation is a task of segmenting out object instances from videos. It has attracted increasing research interests due to its wide potential applications such as video surveillance [1], autonomous driving, and action detection. Based on the level of human supervision in the testing phase, video object segmentation could be roughly classified into supervised (interactive) [2], semi-supervised [3], [4] and unsupervised [5], [6], [7] settings. Among these settings, semi-supervised video object segmentation is a task of propagating instance masks given in the first frame to the entire video, which is a more realistic setting to balance the labor load of human annotation and segmentation accuracy. Semi-supervised video object segmentation is a challenging task. In realistic videos, there exist problems of distinct variations of object scales, fast movement, frequent object disappearance/re-appearance, and heavy occlusions. These problems dramatically reduce the mask propagation performance. The existing works on semi-supervised object segmentation could be roughly classified into two categories: temporal motion-based approaches and spatial cues based approach. Temporal motion-based approaches [8], [9], [10], [11] mainly rely on the temporal continuity information to track the annotations from the first frame through the video sequence. However, temporal motion-based methods are vulnerable to the temporal discontinuity caused by object occlusions and object disappearance. Consequently, the tracking failures are easily propagated to the subsequent frames. Spatial cues based approaches [3], [12], [13], [14], [15], [16] mainly rely on the appearance cues of the target object in the annotated frames to search similar objects in the unannotated frames. Such spatial cues are stable for the disappearance/reappearance of objects but are vulnerable to object deformation and appearance change.

Currently, most existing video object segmentation works rely on time-consuming techniques to improve accuracy. Some works [3], [12], [13], [17], [18] perform fine-tuning with the annotated masks in the testing phase, which introduce additional workload of model training. Some works [19], [20] rely on optical flow, which is quite time-consuming to extract, to represent pixel-level temporal continuities accurately. Some works [3], [18], [21] utilize slow post-processing techniques such as denseCRF to generate masks that coincide well with the object boundaries. Although these mentioned techniques can help increase the segmentation accuracy, they hinder their practical applications to fast/online video segmentation. It is still an open problem of balancing segmentation accuracy and computation efficiency in video object segmentation.

In this paper, we propose a novel Guided Co-Segmentation (GCSeg) network for online semi-supervised object segmentation. Our GCSeg network efficiently combines both motion and static cues to incorporate short-term, middle-term, and long-term temporal inter-frame relationships.

In video segmentation tasks, there exist problems of significant variations of object scales, fast movement, frequent disappearance and re-appearance of objects and heavy occlusions. In order to tackle these problems, we aim to explore the invariant information of the target objects in the video. We assume that such invariant information exists among different pairs of frames with different temporal intervals. The temporal intervals could be ranged from zero to the whole video length. Thus, we are inspired to propose a video segmentation pipeline that could mine different inter-frame relationships (short, middle, and long-term temporal intervals) with the same module.

The short-term relationship refers to the inter-frame relationship between the current frame and the very last previous frame, which processes the most accurate motion cues

W. Liu is with School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore 639798 (e-mail: weide001@e.ntu.edu.sg).

G. Lin is with School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore 639798 (e-mail: gslin@ntu.edu.sg).

T. Zhang is with the Institute for Infocomm Research (I^2R)-Agency for Science, Technology and Research (A*star), Singapore, 138632 (e-mail: Zhang_Tianyi@i2r.a-star.edu.sg).

Z. Liu is with the Nanyang Technological University (NTU), Singapore 639798 (e-mail: zliu016@e.ntu.edu.sg).

Corresponding author: Guosheng Lin.

for propagating previously predicted masks. The middle-term relationship refers to the inter-frame relationship between the current and the previous frames within reasonable short temporal intervals. Such relationships assume reasonable small appearance differences of target objects and relatively significant differences of background, which helps segment out the common target objects. The long-term relationship refers to the relationship between the current frame and the annotated first frame. We aim to mine out the target objects which are close to the ground truth annotation through the whole video. Our approach to incorporate short-term, middle-term, and long-term inter-frame relationships is intuitive and straightforward. When a person searches target objects in video frames, he can either focus on what has been detected in the most similar frame or focus on what are the common objects that exist in other background scenes or focus on which is the same object as the reference groundtruth.

Semi-supervised object segmentation often suffers from the risk of propagating previous errors from predicted masks to subsequent frames. In our work, we propose an adaptive search strategy to decide whether to propagate from the previous predicted mask or not to alleviate this error propagation problem.

We summarize our contributions as follows:

- We propose a GCSeg network which efficiently incorporates short-term, middle-term and long-term temporal inter-frame relationships for semi-supervised object segmentation methods.
- We propose an Adaptive search strategy to alleviate the problem of error propagation.
- Our method achieves state-of-the-art online/real-time semi-supervised object segmentation results on the challenging benchmarks of Davis-2016 and Davis-2017 benchmarks.

II. Related Work

A. Semi-supervised video segmentation

Semi-supervised video object segmentation aims to propagate the mask from the first annotated frame to the rest of the video. The early approaches rely on hand-crafted features to tackle this problem, such as super-pixel propagation [22], object proposals [23] and bilateral space [24]. Recently Deep Neural Networks (DNN) [17], [3], [13], [20], [19], [11] has dramatically increased the performance of semi-supervised video object segmentation. In this section we give a brief review on the DNN-based semi-supervised video object segmentation methods. The methods are roughly categorised as spatial cues based approaches and temporal motion based approaches.

The spatial cues based methods [3], [17], [11], [9], [18], [25], [26] usually apply Fully Convolutional Network (FCN) [2] to capture spatial properties within each video frame to mine out the spatial features that are similar to the groundtruth annotation. In order to mine out the similarity features, model fine-tuning is always applied in the testing phase over the annotated frames. Consequently it introduces additional training complexity. In order to make the prediction conicide well with the object boundary, time-consuming post-processing techniques (e.g., Conditional Random Field (CRF) [27] and contour snapping [3]) are usually applied to refined the segmentation results. Our GCSeg network achieves competitive results with those methods but follows a more efficient pipeline without fine-tuning and post-processing process.

Temporal motion methods [17], [11] propagate the current frame mask to the next frame based on the temporal continuity information. However, such methods are vulnerable to the temporal discontinuity caused by object reappearance or object occlusion. To alleviate the temporal discontinuity problem, Cheng *et al.* [19] and tokmakov *et al.* [20] extract optical flow to capture the accurate pixel-wise temporal information. Li *et al.* [28] rely on re-identification (ReID) module to compensate the error propagation caused by temporal discontinuity. The optical flow is time-consuming to extract, and the ReID makes the pipeline complex to implement. In our work, we propose a much more efficient pipeline to incorporate temporal information.

B. Semantic segmentation

Semantic segmentation is a fundamental computer vision task that aims to assign the correct labels to each pixel in the images or video frames. Fully convolutional networks (FCN) is always applied to pixel-wise prediction tasks. Encoder-Decoder structure [29], [2], [30], [31], [32], [33], [34], [35] is one of the most widely used FCN structure which aims to generate high-resolution prediction maps. Typically the encoder outputs the high-level feature representation of large field-of-view. Such features are of low-resolution which are not suitable for accurate dense prediction tasks. The Decoder aims to recover the high-resolution information from the output feature of the Encoder module. Noh *et al.* [36] upsamples the low-resolution feature maps with a learnable de-convolutional decoder. Dilated convolution [37] is often used to increase the feature map resolution. Skip connections [30] fuse different levels of features for better feature representation. Our network also follows the similar encoder-decoder structure in which the encoder aims to mine the similarity with the reference frame while the decoder aims to recover the high-resolution details.

C. Object co-segmentation

The object co-segmentation task is defined as jointly segmenting similar objects in multiple images based on the assumption of the existence of common objects and distinct backgrounds among these images. In the early works of object co-segmentation works, Vicente *et al.* [38] rely on object recognition method with powerful features extractor, Joulin *et al.* [39] used an efficient convex quadratic energy approximation pipeline, Chiu *et al.* [40] extracted multiple foreground objects by a non-parametric Bayesian model on object proposals and Guo *et al.* [41] extracted the foreground parts by exploiting the common fate exhibiting across different video frames. Recently proposed methods significantly improve performance by adopting CNN [42], [43] to co-segmentation tasks. Compared with other video-processing

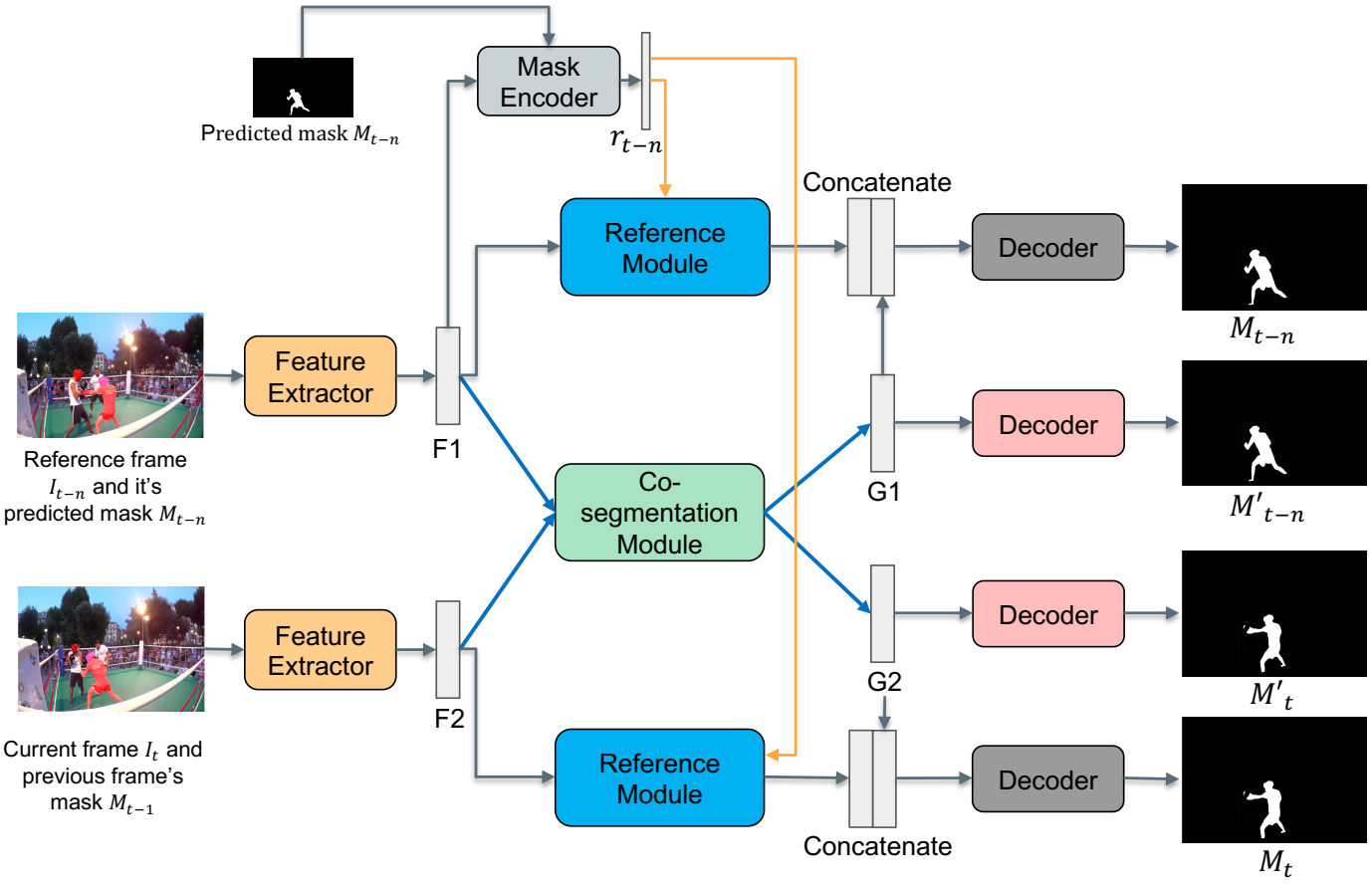

Fig. 1. The pipeline of our GCSeg Network architecture. Our GCSeg Network mainly consists of two modules: Reference Module and Co-segmentation Module. Reference Module is aimed at encoding the foreground region of the reference frame. Co-segmentation Module is aimed at encoding the relationship between the current frame and the previous frames to capture the short, middle and long term relationship. M_t denotes the final prediction of the t_{th} frame, M'_t denotes the prediction from co-segmentation net of the t_{th} frame.

techniques such as spatial-pixel matching and temporal mask propagation, object co-segmentation is more stable to the problem of object appearance variation, shape deformation, and fast motion. The works in [44], [45] apply co-segmentation technique on video segmentation. We incorporate the co-segmentation technique with novel architecture design into the semi-supervised video segmentation task.

D. Few-shot learning

Few-shot learning refers to learning from just a few training examples per class to generalize well to new data. It was first approached by learning how to learn [46] strategy. Few-shot learning received more attention with the developments of DCNN. Mishra *et al.* [47] utilizes neural networks with memory capacities to approach few-shot learning problems. Bertinetto*et al.* [48] utilize fine-tuning technique to predict the model parameter. Metric learning based [49], [50], [51] approaches achieve impressive results on few-shot classification tasks. Our work utilizes a deep metric learning embedding module to generate the foreground features to adapt to new video frames efficiently.

TABLE I
ACRONYMS

Abbreviation	**Meaning**
M_t	The t_{th} frame prediction
M_0	First frame Annotation
I_t	The t_{th} frame
F	Output Features from Feature Extractor module
f	Middle features in co-segmentation net
G	Output Features from co-segmentation module
r_{t-n}	Masked Foreground Reference Feature Vector
GT_t	The ground truth of t_{th} frame
R	The output features of Reference Module
$R1$	The features from upper stream of Reference Module
$R2$	The features from bottom stream of Reference Module
$D1$	Similarity map
$D1'$	Normalized similarity map

III. METHOD

As illustrated in Figure 1, Our GCSeg network is mainly composed of two parts: Reference Module and Co-segmentation Module. A video is represented by a frame sequence $I = \{I_0, I_1, \cdots, I_t \cdots I_T\}$. Reference Module is aimed at encoding the foreground regions of reference frame I_{t-n} to retrieve similar features in other feature space. The Co-segmentation Module extracts regional representation from

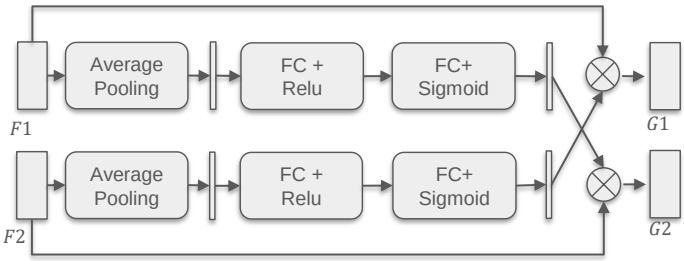

Fig. 2. The co-segmentation module aims to mine out the common objects existing both in the current t_{th} frame and $(t-n)_{th}$ frame. $F1$ and $F2$ denote the output of Feature Extractor module to represent the feature of input frame and predicted mask. $G1$ and $G2$ denote the output feature which encodes the similarity between the input feature pair.

the current video frame I_t and the reference frame I_{t-n} within the whole video sequence. In our GCSeg, Reference Module guides Co-segmentation Module in order to generate a more informative frame feature to obtain more accurate segmentation results. In this section, we introduce our GCSeg network and how it can incorporate short-term, middle-term, and long-term temporal relationships. Then we describe our Adaptive Search Strategy and how it adaptive copes with the error propagation problem. For convenience, we denote the provided ground truth of the first frame as M_0, and the other t_{th} predicted masks as M_t.

A. GCSeg Network

In this section, we first briefly introduce our Training and Inference Procedure. Then we introduce the Feature Extractor module, which aims to encode the feature of the input frames. Next, we introduce the main parts of our GCSeg network: Reference Module and Co-segmentation Module. Finally, we describe the Decoder, which aims to produce the final segmentation outputs.

1) Training and Inference Procedure: In this section, we introduce how to sample our reference frames during the training and inference phase. A video clip from the first to current frame is represented as $I = \{I_0, I_1, \cdots, I_t\}$. During the training procedure, we randomly select a mini-batch of videos as training frames. For each video, we randomly select one pair of frames in each video. One frame serves as the reference frame, and the other frame serves as the target frame to be segmented. Our proposed GCSeg network focus on encoding the relationships between the reference frame and the target frame. Based on the temporal interval between the reference and target frame, the relationship could be roughly classified as short-term, middle-term, and long-term temporal relationships.

In the inference phase, we evenly split the video clip $I = \{I_0, I_1, \cdots, I_t\}$ into *n* temporal fragments $s_1, s_2, \cdots, s_n$. To represent a different temporal relationship, we randomly select one frame from each fragment and add it to the reference frames. We also add the annotated first frame into the set of reference frames, which encloses the groundtruth reference information. Each reference frame generates segmentation prediction on the target frame. The final segmentation prediction on the target frame is calculated as the average of all the results generated with different reference frames.

2) Feature Extractor: In this section, we introduce how to encode the input frames and masks (either predicted masks or ground truth) into the feature space. We use I_t to denote the input t_{th} frame to segment, M_t to denote the predicted binary mask of t_{th} frame. M_0 denotes the ground-truth mask of the first annotated frame.

Our Feature Extractor takes I_t (*resp.*,I_0 for annotated frame) and M_{t-1} (*resp.*, M_0 for annotated frame) as inputs and outputs $\mathbf{F}_t \in \mathbb{R}^{W \times H \times C}$ as the encoded feature of the input frames and masks. Here we use W,H,C to denote the width, height, and dimension (number of channels) of the feature maps.

Our Feature Extractor is mainly based on the backbone of the ResNet101. Similar to RGMP [52], we modify our network to adapt 4 channel input by implanting an additional single channel convolution with the first convolution network from our the backbone, which is depicted as the masked input images in Figure 1. The predicted mask of the previous frame always provides a localization cue of the current frame. We combine the current frame I_t, and it is previously predicted mask M_{t-1} and feed into our Feature Extractor to capture the short term inter-frame relationship. We input I_t and M_{t-1} (or I_{t-n} and M_{t-n}) into our Feature Extractor to generate the encoded feature $\mathbf{F}_t$ (or $\mathbf{F}_{t-n}$). All the Feature Extractor depicted in Figure 1 share the weights and are fixed after pre-training.

3) Co-segmentation Module: In our Co-segmentation Module, we aim to mine out the common objects which exist both in t_{th} frame (current frame) and $(t-n)_{th}$ ($n \in [0, T-1]$, randomly chosen) frame (reference frame), based on the assumption that the frames in different frames usually possess common foreground objects.

We illustrate the structure of our Co-segmentation Module in Figure 2. For simplicity, we denote $F1 = \mathbf{F}_t$ and $F2 = \mathbf{F}_{t-n}$. Our Co-segmentation Module takes $F1$ and $F2$ as inputs and outputs $G1$ and $G2$. The output feature encodes the similarity between the current t_{th} frame and $(t-n)_{th}$ frame. It is worth noticing that we encode short-term, middle-term, and long-term inter-frame relationships with such similarity since the reference frame is randomly chosen from the previous frames with a broad range of time intervals.

The structure of our Co-segmentation Module is described as follows: first we input $F1$ (*resp.*, $F2$) into an average pooling and fully connected layer to obtain $f1$ (*resp.*,$f2$). The

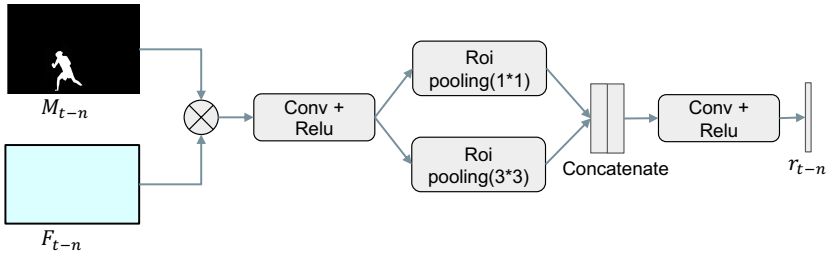

Fig. 3. The mask encoder aims to generate reference object feature r_{t-n} to represent the reference frame I_{t-n} and its groundtruth or predicted mask M_{t-n}. F_{t-n} denotes the output of Feature Extractor module.

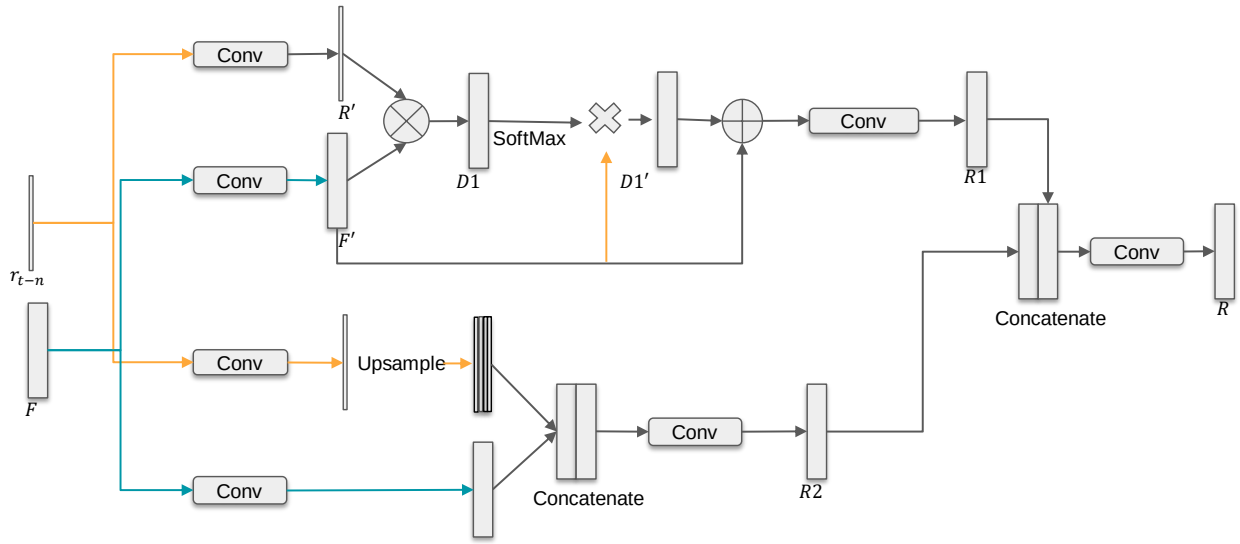

Fig. 4. The Reference Module. Our reference module utilizes r_{t-n} as a reference feature to mine similar feature in F ($F1$ and $F2$, the output of of Feature Extractor) , and output R as the final representation.

output feature is calculated by $G1 = F1 \cdot f2$ and $G2 = F2 \cdot f1$, in which $\cdot$ denotes the operation of broad-cast element-wise multiplication between the input feature vector and the feature vector at each spatial position of the input feature matrix.

The results of $G1$ and $G2$ incorporate the similarity information between the input feature maps corresponding to two frames. Another explanation is that $f1$ (*resp.*, $f2$) is an attention from another frame feature $F1$ (*resp.*, $F2$), and $G2$ (*resp.*,$G1$) is the attended feature which highlight the feature which is similar to the objects in another frame. For implementation details, we input different levels of output feature maps of Feature Extractor into our Co-segmentation Module. We use different levels of features base on the following consideration: lower level features are more precise on the spatial localization while the higher-level features contain more semantic information. We concatenate different levels of feature maps from the co-segmentation module as the final generated feature G (For simplicity, we utilize G to denote $G1$ and $G2$ for short).

4) Reference Module: Our Reference Module is aimed to mine out the features which are similar to the reference features. The reference feature is generated by our mask encoder, which encodes the foreground property of the reference frame.

In this section, we first introduce how to generate reference feature r_{t-n} using mask encoder and then describe how to mine out the features using the Reference Module.

(a) Mask Encoder Our Mask Encoder takes inputs of $\mathbf{F}_{n-t}$ and M_{n-t} into a Mask Encoder to generate r_{n-t} as the reference feature of the reference frame foreground mask. We illustrate the structure of Mask encoder in Figure 3. Our Mask encoder is built as follows: we first element-wise multiply $\mathbf{F}_{t-n}$ with M_{t-n} to zero-out background features. Then we perform pyramid ROI pooling based on the predicted mask bounding box of M_{t-n}. Specifically, we divide the bounding box into 3×3 grid and perform ROI pooling for each gird and the whole bounding box region. In this way, we get the features of not only the whole objects but also objects parts. We perform such pyramid pooling in order to tackle the problem of object occlusion and deformation. Normally we could identify the whole objects by mining the similar whole object region. When an occlusion occurs, we are still able to identify the object by mining similar parts.

To decide whether to use object regions or part regions as the reference features, we adaptively learn which feature to emphasize. Specifically, We concatenate all 10 ROI-pooled features to obtain feature map of $10 \times C$, and input the concatenated feature into a 10×1 convolutional layer to obtain a $1 \times C$ feature as the final representation feature r_{t-n}. Here we use C to denote the dimension (number of channels) of the feature.

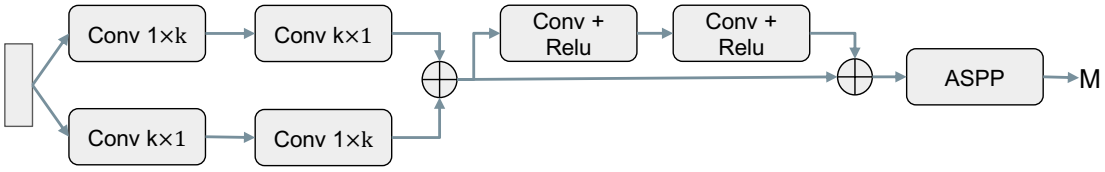

Fig. 5. The Decoder Module to predict the final prediction M.

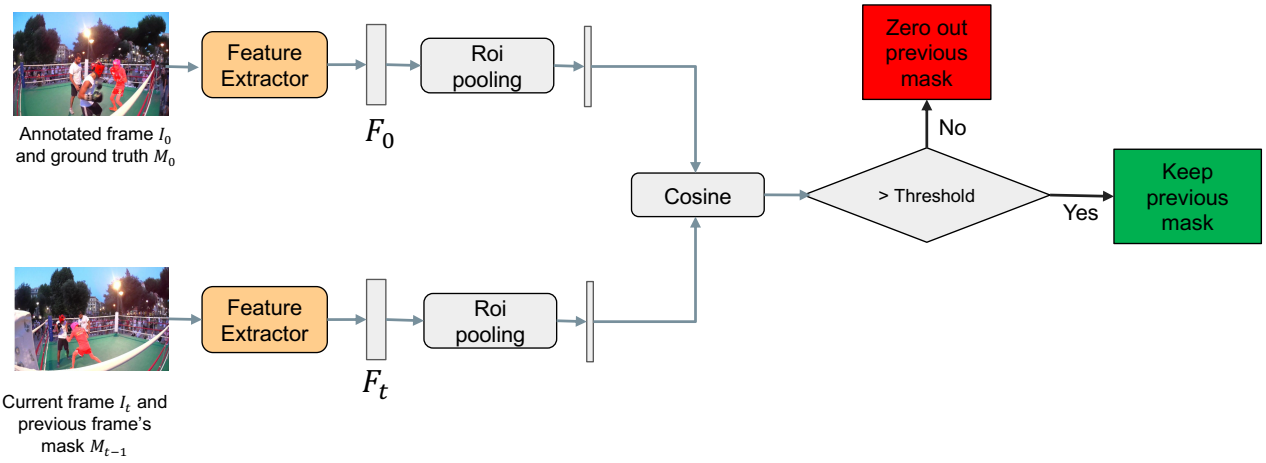

Fig. 6. The judgement module in our Adaptive Search, aiming to decide whether the predicted mask of the last previous frame M_{t-1} is reliable enough or not.

(b) Reference Module The structure of our Reference Module is illustrated in Figure 4. Given the features $F1$ and features $F2$ generated by the Feature Extractor, and the reference feature r_{t-n} generated by Mask Encoder, we aim to encode similarity between reference feature r_{t-n} and features $F1$ (*resp.* $F2$) by our reference module and output the features R_t (*resp.* R_{t-n}).

As is depicted in Figure 1, the Reference Module for F_1 and F_2 are of the same structure with the same weight. For simplicity, we disregard the frame indices and denote the input as F and the output as R. The reference module is illustrated in Figure 4, we encode the similarity between the features r_{t-n} and F with a two-stream structure and combine outputs of the two-stream as the final similarity features. In the upper stream, we first input F and r_{t-n} into 1×1 convolutional layer to generate F' and R'. Such an operation aims to map the features into the same space. The output feature $R1$ is calculated as follows: We first incorporate the similarity between R' and F' with $D1 = R' * F'$, where $*$ denotes matrix multiplication. Next, we perform spatial normalization to generate attention feature $D1'$. We apply the attention to the $D1$ to generate attended feature $D1$ to highlight the most confident feature. This process is described in Equation 1 and Equation 2 as follows:

$$f^{ijc} = \frac{\exp({f_a}^{ijc})}{\sum_{i,j} \exp({f_a}^{ijc})}, \tag{1}$$

$${f_b}^{ijc} = f^{ijc} {f_a}^{ijc}. \tag{2}$$

Here, i,j and c denote the spatial index corresponding to height H,width W and dimension C (number of channels) of the feature maps. ${f_a}^{ijc}$ refers to gird value of $D1$ at corresponding value, and ${f_b}^{ijc}$ refers to gird value of $D1'$ at corresponding value. Finally, the output feature of our upper Reference Module $R1$ is calculated by $R1 = F' + D1'$, which takes both the original feature and the attended feature into consideration. In the bottom stream, we mine out the similarity between feature r_{t-n} and F with a different but simple way. We resize and concatenate the feature map r_{t-n} and F, and input the concatenated feature maps into a convolutional block to generate similarity feature $R2$. Finally, we concatenate $R1$ and $R2$ and input the concatenated features into a convolution block to generate the final similarity features R.

5) Decoder: In our Decoder Module, we aim to decode R generated by Reference Module and G generated by Co-segmentation Module into predicted segmentation mask M_t. The structure of the Decoder is illustrated in Figure 5.

Our Decoder sequentially consists of three parts: a global convolution block [53], a residual block [54] without batch normalization and ASPP [55] Block. Global convolution block and ASPP block are aimed to enlarge the receptive field to overcome the locality limitation of convolution operations. As illustrated in Figure 5, we concatenate feature G and R and input it into our Decoder and a channel-wise softmax layer to obtain the predicted mask M_t and M_{t-n}. Based on the ground-truth mask GT_t and GT_{t-n} The segmentation loss is formulated as Equation 3 .

$$\begin{aligned} Loss = {} & L_{bce}(M_t, GT_t) + L_{bce}(M_{t-n}, GT_{t-n}) \\ & + L_{bce}(M_t^{'}, GT_t) + L_{bce}(M_{t-n}^{'}, GT_{t-n}), \end{aligned} \tag{3}$$

where L_{bce} denotes sum of pixel-wise Binary Cross Entropy loss between the predicted mask and corresponding

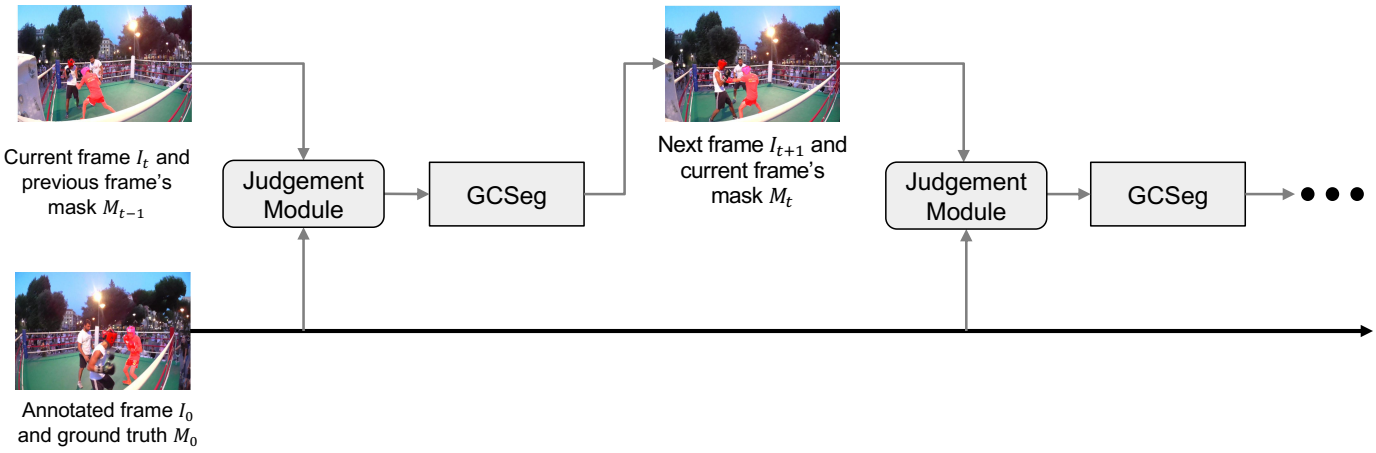

Fig. 7. We use the judgement module to estimate the reliability of the predicted mask. It helps to decide whether the previous prediction mask can be used or not in the GCSeg Network.

groundtruth mask. As illustrated in Figure 1, $M_t^{'}$ and $M_{t-n}^{'}$ are the result masks of sequentially inputting feature $G1$ and $G2$ into Decoder and and channel-wise softmax layer. M_t (or M_{t-n}) is the result mask of concatenating G_1 (or G_2) with features R_{t-n} and sequentially being input into Decoder and softmax layer.

B. Adaptive Search Strategy

One of the most challenging problems of video object segmentation is the object re-appearance in the current t_{th} frame caused by the object occlusion or disappearance in the previous frames. In such situation, the predicted masks of the last previous frame, which is denoted as M_{t-1}, is misleading for the segmentation for t_{th} frame, since there is no target object is the $(t-1)_{th}$ frame and errors will be propagated in the latter frames. Thus, we propose an Adaptive Search strategy to tackle this error propagation problem.

In our Adaptive Search Strategy, we first utilize a Judgment Module to decide whether M_{t-1} is reliable enough based on the reference features. If M_{t-1} is reliable, we keep the last previous frame predicted mask as input M_{t-1} in the co-segmentation stream. Otherwise, we set M_{t-1} as the zero-mask, which means our feature extractor only encodes the feature of the current frame I_t. Consequently, the network discards the short-term relationship and relies on middle-term and long-term relationships to guide the segmentation of the current frame.

Judgement Module is aimed to decide whether the last previous predict mask M_{t-1} is reliable enough. Given the feature F_0 (*resp.*,F_t) and the mask M_0 (*resp.*,M_{t-1}), we perform ROI pooling on F_0 (*resp.*, F_t) over the bounding box region of M_0 (*resp.*,M_{t-1}). Then we calculate cosine similarity between these two ROI-pooled features. If the similarity score is higher than some threshold (*e.g.*, 0.7), we decide that M_{t-1} is confident enough. Otherwise, M_{t-1} is not reliable and should be zeroed-out. During training, we random zero out the previous mask with a possibility of 0.5 to simulate the non-confident situation.

IV. Experiments

A. Experimental Set-up

We perform our experiments on Davis-2016 [56], Davis-2017 [57] and YoutubeObject [22] datasets. Davis-2016 contains 30 training videos and 20 validation videos. Each of the videos has single-instance pixel-wise groundtruth annotations. Davis-2017 is a more complicated and challenging dataset. It contains 60 training videos and 30 validation videos with multiple instance cases and frequent occurrence of object occlusion, object disappearance/reappearance, camera motion, etc. Our pipeline is intended for binary segmentation for every single instance. To cope with the multi-instance case of Davis-2017, we input the annotation of every single instance independently into the model for mask propagation and merge the instance binary mask with an argmax operation. Since the YoutubeObject dataset does not have splits of training and testing data, we do not train on the YoutubeObject dataset. We apply the model trained on Davis-2017 to evaluate YoutubeObject.

Following [56], we use the same evaluation metrics to measure the segmentation performance in terms of intersection over union ($\mathcal{J}$) and contour accuracy ($\mathcal{F}$). We also report the mean of $\mathcal{J}$ and $\mathcal{F}$ metrics, which is denoted as $\mathcal{J}\&\mathcal{F}$.

Following [52], we pre-train our model first on the simulated augmented datasets and then perform finetuning on the target video segmentation training datasets. We perform data augmentation by generating simulated video datasets from other static images datasets (*i.e.*, Pascal VOC [58] and MSRA10K [59]). We generate the augmented dataset in two ways. Way-1: we randomly scale/deform the foreground image and paste into the same background with a small shift to simulate the case of motion between adjacent frames. Way-2: we randomly scale/deform the foreground and paste into a different background to simulate the case of fast motion and object occlusion. The examples of this simulated dataset are illustrated in Figure 8.

To simulate the case of error accumulation over time, we finetune our model on video frames in a recurrence way similar to MaskRNN [9]. Specifically, we use the predicted

output from the last frame as a guidance score map for the current frame. Note that all the operations in our network are differentiable, which allows us to perform end-to-end training.

B. Comparison with the State-of-the-art

We report our results and compare with the state-of-the-art methods. We report the results on the datasets of YoutubeObject [27], Davis-2016 [56] and Davis-2017 [57] in Table IV , Table II and Table III, respectively.

Some existing methods usually apply some time-consuming techniques in the prediction stage. These time-consuming techniques can be roughly categorized into 3 types: Finetuning (FT), Optical flow (OF) and Post-processing (PP). Finetuning (FT) refers to finetuning the models using the available annotated frames in the testing phases. Optical flow (OF) means extracting inter-frame optical flow to obtain accurate pixel-wise motion information. Post-processing (PP) means performing computationally expensive refinement post-processing step, such as DenseCRF [27], on the predicted masks. For clear analysis and fair comparison, we also indicate the usage of these test-stage processing techniques in the tables for every comparison method. Kindly note that our method does not apply any of these test-stage processing techniques. We also list the average time for predicting each frame for an accurate comparison of the efficiency.

In these three result tables, the comparison methods are grouped into two parts. The upper part of the tables presents the methods which perform a time-consuming finetuning step in the test stage. Finetuning will lead to additional model training in the inference stage and hinders the efficiency of these methods. The lower parts in these result tables present the methods which do not rely on finetuning step. Such methods are potentially efficient for real-time object video segmentation. Our method belongs to the second category, which does not involve a finetuning step. Since we do not apply the time-consuming techniques such as finetuning, optical flow generation, and post-processing refinement, our method is efficient in the inference stage and could be applied for real-time/online video segmentation purpose.

As is reported in Table II, Table III, and Table IV, we achieve the best performance among the methods with comparable computational efficiency. It is worth noticing that we utilize none of the time-consuming techniques in the inference stage (*i.e.*, finetuning, optical flow, and post-processing) which makes our method as one of the most efficient ones. We also present the qualitative examples of our segmentation results in Figure 9.

C. Ablation study

1) Components of GCSeg Network: We conduct ablation studies on Davis-2016 [56] to evaluate the contribution of different components of our network. We report the ablation results on Davis-2016 in Table V.

Our baseline model refers to the setting of removing both the Reference Module and Co-segmentation Module, which means that we directly use Feature Extractor to encode the current frame and the associated mask and input the resulting features into the Decoder to predict the mask for the current frame. We train the baseline module using the Binary Cross Entropy loss. The results are shown in the first row of Table V.

Next, we add our Reference Module to the baseline module. In this case, we apply our Reference Module directly onto the result features of the Feature Extractor and input into the Decoder to generate the predicted masks. This model is trained with the Binary Cross Entropy loss. This result is shown in the second row of Table V (denoted as "Ref").

Third, we add our Co-segmentation Module to the baseline model without using the Reference Module. In this case, we directly input the feature maps of our Co-segmentation Stream into the Decoder to obtain the predicted mask of the input frame. The results are shown in the third row of Table V (denoted as "Co-seg").

Finally, we add both the Reference Module and Co-segmentation Module to the baseline module and report the results in the last row of Table V. Note the in this table; we

TABLE II
RESULTS ON THE VALIDATION SET OF DAVIS-2016 DATASET. FT: FINE-TUNING WITH THE FIRST FRAME OF THE VIDEO IN THE TESTING PHASE; PP: POST-PROCESSING; OF: OPTICAL FLOW; † : WITHOUT PRE-TRAINING ON OTHER DATASETS; AD: PRE-TRAIN ON ADDITIONAL DATA ; TIME(S): THE AVERAGE TIME (IN SECONDS) SPENT ON PREDICTING EACH FRAME. OUR METHOD OUTPUT-PERFORMS OTHER METHODS WITH COMPARABLE PREDICTION SPEED.

Method	FT	PP	OF	AD	$\mathcal{J}$ (%)	$\mathcal{F}$(%)	$\mathcal{J}\&\mathcal{F}$(%)	Time(s)
MSK [17]	✓	✓	✓	✓	79.7	75.4	77.5	12s
MaskRNN [9]	✓		✓	✓	80.7	80.9	80.8	0.6s
OnAVOS [18]	✓	✓		✓	**86.1**	84.9	85.5	13s
OSVOS [3]	✓	✓		✓	79.8	80.6	80.2	9s
Lucid [11]	✓	✓	✓	✓	84.8	-	-	190s
STCNN[60]	✓	✓	✓	✓	83.8	83.8	83.3	3.9
OSVOSs [12]	✓	✓		✓	85.6	87.5	86.5	4.5s
PReMVOS [21]	✓		✓	✓	85.5	**88.6**	**87.0**	70s
OnAVOS[18]				✓	39.5	–	–	3.78s
BVS [24]					60.0	58.8	59.4	0.37s
OTP [61]				✓	77.9	76.0	76.9	0.6s
PML [62]				✓	79.3	75.5	77.4	0.27s
A-GAME [63]				✓	82.2	82.0	82.1	0.07s
OSMN [64]				✓	74.0	-	-	0.14s
SiamMask[4]				✓	71.7	67.8	69.7	**0.02s**
FEELVOS[8]				✓	81.1	82.2	81.7	0.5s
FAVOS[61]				✓	82.4	79.5	80.8	1.8s
RGMP [52]				✓	81.5	**82.0**	81.7	0.13s
RGMP † [52]					68.6	68.9	68.7	0.13s
Ours †					70.5	71.7	71.1	0.28s
Ours				✓	**82.6**	81.7	**82.2**	0.28s

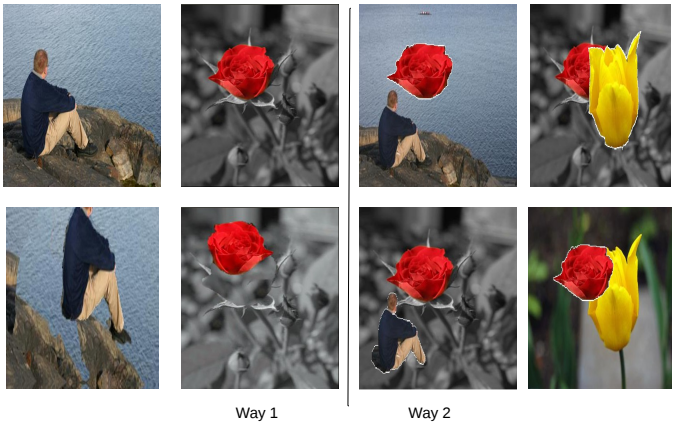

Fig. 8. The example images of the augmented dataset simulated from static images.

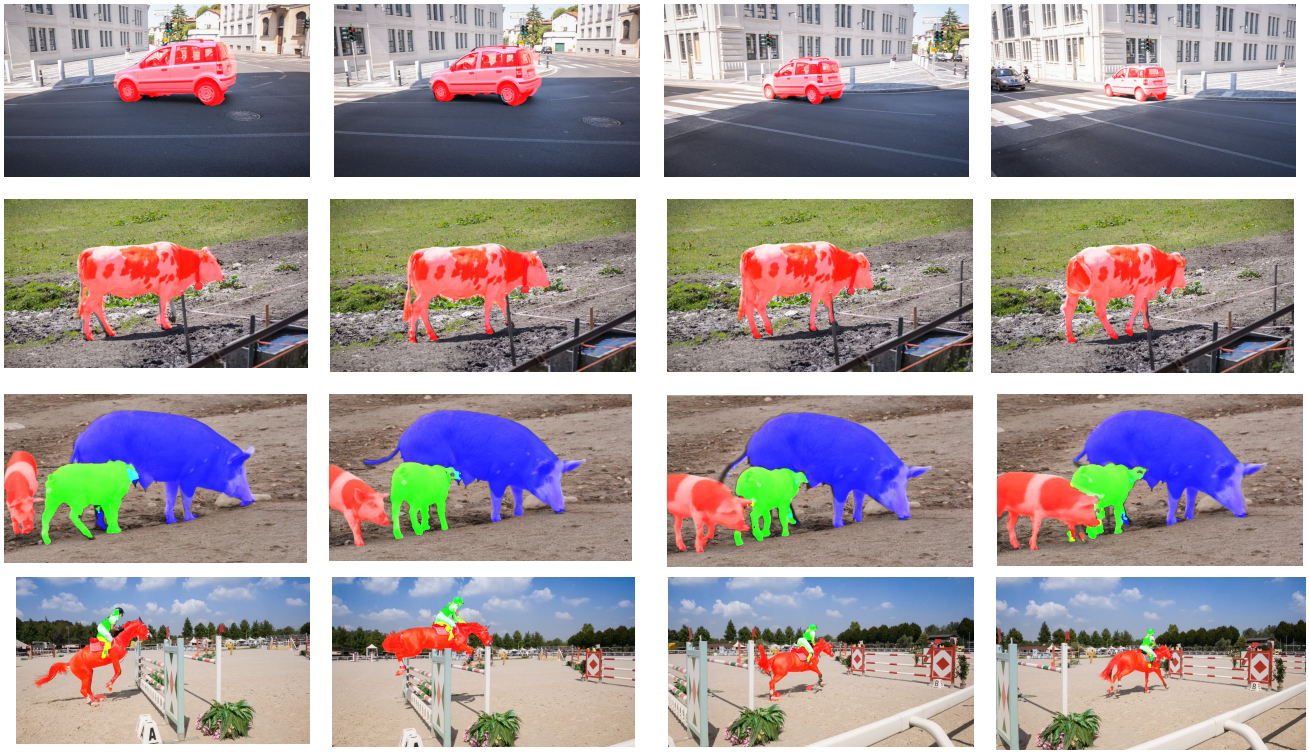

Fig. 9. The qualitative results on DAVIS-2016 (The first two rows) and DAVIS-2017 (The last two rows)

TABLE III
RESULTS ON DAVIS-2017 VALIDATION DATASET. FT: FINE-TUNING ON THE FIRST FRAME OF THE TEST VIDEO; PP: POST-PROCESSING; OF: OPTICAL FLOW; AD: PRE-TRAIN ON ADDITIONAL DATA ; TIME(S): THE AVERAGE TIME (IN SECONDS) SPENT ON PREDICTING EACH FRAME; OUR METHOD OUTPERFORM OTHER METHODS WITH COMPARABLE PREDICTION SPEED.

Method	FT	PP	OF	AD	$\mathcal{J}(\%)$	$\mathcal{F}(\%)$	$\mathcal{J}\&\mathcal{F}(\%)$	Time(s)
MaskRNN [9]	✓		✓	✓	60.5	–	–	9s
OSMN [64]	✓	✓		✓	60.8	–	–	–
OnAVOS [18]	✓	✓		✓	61.6	69.1	65.3	13s
OnAVOS-Ensemble [18]	✓	✓		✓	64.5	71.2	67.8	30s
VideoMatch [65]	✓			✓	61.4	–	–	2.62s
ReID [28]	✓		✓	✓	67.3	71.0	69.1	2.33s
OSVOS^S [12]	✓	✓		✓	64.7	71.3	68.0	–
CINM† [66]	✓	✓	✓	✓	67.2	74.4	70.7	$\sim$ 108s
PReMVOS† [21]	✓		✓	✓	**74.3**	**82.2**	**78.2**	$\sim$ 70s
OnAVOS [18]				✓	39.5	–	–	3.78s
MaskRNN [9]			✓	✓	45.5	–	–	0.6s
VideoMatch [65]				✓	56.5	–	–	0.35s
RGMP [52]				✓	64.8	**68.6**	66.7	–
Ours				✓	**65.5**	68.4	**67.0**	0.92s

TABLE IV
RESULTS ON YOUTUBEOBJECT DATASET.PERFORMANCE MEASURED IN THE MEAN OF $\mathcal{J}$. FT: FINE-TUNING ON THE VIDEO WITH THE FIRST FRAME OF THE TEST VIDEO; OF: OPTICAL FLOW; PP: POST-PROCESSING; AD: PRE-TRAIN ON ADDITIONAL DATA; TIME(S): THE AVERAGE TIME (IN SECONDS) SPENT ON PREDICTING EACH FRAME. OUR METHOD PERFORMS THE BEST COMPARED TO THOSE METHODS WITH COMPARABLE PREDICTION SPEED.

Method	YoutubeObjs	FT	OF	PP	AD	Speed(s)
MaskTrack [17]	71.7	✓			✓	12s
OSVOS [3]	**74.1**	✓			✓	10s
OFL [67]	67.5	-	✓	✓		42.2s
BVS [24]	58.4	-	✓			0.37s
MaskTrack-B [17]	66.5					0.24s
OSVOS-B [3]	44.7					0.14s
OSNM [64]	69.0				✓	0.14s
Ours	**71.2**				✓	0.19s

do not pre-train the networks on other augmented datasets for ablation analysis. Thus, the results shown here are different from that in the results Table II.

We observe that using a single component, either Reference Module or Co-Segmentation Module improves the segmentation results over the baseline. It demonstrates the effectiveness of each component. Moreover, adding both of the components could further increase the performance, which indicates that the Reference Module and Co-Segmentation Module capture complementary information to improve the system performance.

2) Adaptive Search: We perform ablation study of our Adaptive Search strategy on Davis-2017 [57], since this dataset has more frequent occurrences of object occlusion and disappearance/reappearance cases. We report the results without and with the Adaptive search strategy in Table VI. We observe that using Adaptive Search helps to improve the segmentation results.

3) Data augmentation: Applying data augmentation is a common practice in recent video segmentation methods. As is introduced in Sec. IV-A, we follow the method in [52] to generate simulated datasets for data augmentation. We perform ablation studies on Davis-2016 to evaluate the effectiveness of pre-training the model on augmented simulated datasets. The results are reported in Table VII, which shows that such data augmentation helps to improve the segmentation performance.

TABLE V
ABLATION STUDY ON DAVIS-2016 VALIDATION DATASET OF EACH NETWORK COMPONENT. REF: USING REFERENCE MODULE ONLY. CO-SEG: USING CO-SEGMENTATION MODULE ONLY. REF + CO-SEG: USING BOTH REFERENCE AND CO-SEGMENTATION MODULES. IT CLEARLY SHOWS THAT EACH STEAM CLEARLY IMPROVES THE PERFORMANCE AND COMBINING THEM FURTHER IMPROVE THE PERFORMANCE.

Method	$\mathcal{J}$ (%)	$\mathcal{F}$(%)	$\mathcal{J}\&\mathcal{F}$(%)
baseline	55.2	51.3	53.2
Ref (ours)	62.5	61.2	61.9
Co-seg (ours)	61.1	59.7	60.4
Ref + Co-seg (ours)	70.5	71.7	71.1

TABLE VI
ABLATION STUDY ON ADAPTIVE SEARCH ON DAVIS-2017 VALIDATION DATASET.

-	$\mathcal{J}$ (%)	$\mathcal{F}$(%)	$\mathcal{J}\&\mathcal{F}$(%)
Adaptive Search off	65.1	68.0	66.6
Adaptive Search on	65.5	68.4	67.0

V. CONCLUSION

In this paper, we propose a novel approach for instance-level semi-supervised video object segmentation which incorporates short-term, middle-term, and long-term temporal inter-frame relationships. We demonstrate that our GCSeg network achieves state-of-the-art performance without using time-consuming techniques in the prediction stage. Thus, our approach is applicable for realtime video object segmentation tasks since it achieves a balance between the segmentation accuracy and prediction efficiency.

ACKNOWLEDGMENT

This research is supported by the National Research Foundation Singapore under its AI Singapore Programme (Award Number: AISG-RP-2018-003), the NTU start-up grant, and the MOE Tier-1 research grants: RG126/17 (S), RG28/18 (S) and RG22/19 (S).

REFERENCES

[1] L. Zhao, Z. He, W. Cao, and D. Zhao, "Real-time moving object segmentation and classification from hevc compressed surveillance video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 6, pp. 1346–1357, 2016.

[2] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[3] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool, "One-shot video object segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 221–230.

[4] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. Torr, "Fast online object tracking and segmentation: A unifying approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1328–1338.

[5] B. Luo, H. Li, F. Meng, Q. Wu, and K. N. Ngan, "An unsupervised method to extract video object via complexity awareness and object local parts," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 7, pp. 1580–1594, 2017.

TABLE VII
ABLATION STUDY ON DATA AUGMENTATION FROM STATIC IMAGES ON DAVIS-2016 VALIDATION DATASET. WE FOLLOW THE METHOD IN [17] TO PERFORM DATA AUGMENTATION.

-	$\mathcal{J}$ (%)	$\mathcal{F}$(%)	$\mathcal{J}\&\mathcal{F}$(%)
w/o aug	70.5	71.7	71.1
w/ aug	82.6	81.7	82.2

[6] M. Rubinstein, A. Joulin, J. Kopf, and C. Liu, "Unsupervised joint object discovery and segmentation in internet images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 1939–1946.

[7] J. Zhang, J. Yu, and D. Tao, "Local deep-feature alignment for unsupervised dimension reduction," *IEEE transactions on image processing*, vol. 27, no. 5, pp. 2420–2432, 2018.

[8] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9481–9490.

[9] Y.-T. Hu, J.-B. Huang, and A. Schwing, "Maskrnn: Instance level video object segmentation," in *Advances in Neural Information Processing Systems*, 2017, pp. 325–334.

[10] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017, pp. 2980–2988.

[11] A. Khoreva, R. Benenson, E. Ilg, T. Brox, and B. Schiele, "Lucid data dreaming for multiple object tracking," *arXiv preprint arXiv:1703.09554*, 2017.

[12] 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," *arXiv preprint arXiv:1709.06031*, 2017.

[13] J. Shin Yoon, F. Rameau, J. Kim, S. Lee, S. Shin, and I. So Kweon, "Pixel-level matching for video object segmentation using convolutional neural networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2167–2176.

[14] Y. Wang, W. Ding, B. Zhang, H. Li, and S. Liu, "Superpixel labeling priors and mrf for aerial video segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[15] D. Giordano, F. Murabito, S. Palazzo, and C. Spampinato, "Superpixel-based video object segmentation using perceptual organization and location prior," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4814–4822.

[16] F. Galasso, R. Cipolla, and B. Schiele, "Video segmentation with superpixels," in *Asian conference on computer vision*. Springer, 2012, pp. 760–774.

[17] F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung, "Learning video object segmentation from static images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2663–2672.

[18] P. Voigtlaender and B. Leibe, "Online adaptation of convolutional neural networks for video object segmentation," *arXiv preprint arXiv:1706.09364*, 2017.

[19] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang, "Segflow: Joint learning for video object segmentation and optical flow," in *Proceedings of the IEEE international conference on computer vision*, 2017.

[20] P. Tokmakov, K. Alahari, and C. Schmid, "Learning video object segmentation with visual memory," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[21] J. Luiten, P. Voigtlaender, and B. Leibe, "Premvos: Proposal-generation, refinement and merging for video object segmentation," *arXiv preprint arXiv:1807.09190*, 2018.

[22] S. D. Jain and K. Grauman, "Supervoxel-consistent foreground propagation in video," in *European Conference on Computer Vision*. Springer, 2014, pp. 656–671.

[23] F. Perazzi, O. Wang, M. Gross, and A. Sorkine-Hornung, "Fully connected object proposals for video segmentation," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3227–3234.

[24] N. Märki, F. Perazzi, O. Wang, and A. Sorkine-Hornung, "Bilateral space video segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 743–751.

[25] J. Jiang and X. Song, "An optimized higher order crf for automated labeling and segmentation of video objects," *IEEE Transactions on*

Circuits and Systems for Video Technology, vol. 26, no. 3, pp. 506–516, 2015.

[26] B. Wang, Z. Fu, H. Xiong, and Y. F. Zheng, "Transductive video segmentation on tree-structured model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 5, pp. 992–1005, 2016.

[27] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," in *Advances in neural information processing systems*, 2011, pp. 109–117.

[28] X. Li and C. Change Loy, "Video object segmentation with joint re-identification and attention-aware mask propagation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 90–105.

[29] G. Lin, A. Milan, C. Shen, and I. D. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation." in *CVPR*, vol. 1, no. 2, 2017, p. 5.

[30] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.

[31] W. Liu, C. Zhang, G. Lin, and F. Liu, "Crnet: Cross-reference networks for few-shot segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4165–4173.

[32] J. Wang, W. Jiang, L. Ma, W. Liu, and Y. Xu, "Bidirectional attentive fusion with context gating for dense video captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7190–7198.

[33] J. Yu, J. Li, Z. Yu, and Q. Huang, "Multimodal transformer with multi-view visual representation for image captioning," *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[34] L. Zhou, Y. Zhou, J. J. Corso, R. Socher, and C. Xiong, "End-to-end dense video captioning with masked transformer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8739–8748.

[35] Z. Yu, J. Yu, Y. Cui, D. Tao, and Q. Tian, "Deep modular co-attention networks for visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6281–6290.

[36] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1520–1528.

[37] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[38] S. Vicente, C. Rother, and V. Kolmogorov, "Object cosegmentation," in *CVPR 2011*. IEEE, 2011, pp. 2217–2224.

[39] A. Joulin, F. Bach, and J. Ponce, "Multi-class cosegmentation," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 542–549.

[40] W.-C. Chiu and M. Fritz, "Multi-class video co-segmentation with a generative multi-video model," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 321–328.

[41] J. Guo, L.-F. Cheong, and R. Tan, "Video foreground cosegmentation based on common fate," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 3, pp. 586–600, 2016.

[42] H. Chen, Y. Huang, and H. Nakayama, "Semantic aware attention based deep object co-segmentation," *arXiv preprint arXiv:1810.06859*, 2018.

[43] A. Faktor and M. Irani, "Co-segmentation by composition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1297–1304.

[44] H. Fu, D. Xu, B. Zhang, and S. Lin, "Object-based multiple foreground video co-segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3166–3173.

[45] J. C. Rubio, J. Serrat, and A. López, "Video co-segmentation," in *Asian Conference on Computer Vision*. Springer, 2012, pp. 13–24.

[46] J. Schmidhuber, "Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook," Ph.D. dissertation, Technische Universität München, 1987.

[47] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel, "A simple neural attentive meta-learner," *arXiv preprint arXiv:1707.03141*, 2017.

[48] L. Bertinetto, J. F. Henriques, J. Valmadre, P. Torr, and A. Vedaldi, "Learning feed-forward one-shot learners," in *NIPS*, 2016, pp. 523–531.

[49] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *NIPS*, 2017.

[50] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML Deep Learning Workshop*, vol. 2, 2015.

[51] J. Yu, D. Tao, J. Li, and J. Cheng, "Semantic preserving distance metric learning and applications," *Information Sciences*, vol. 281, pp. 674–686, 2014.

[52] S. Wug Oh, J.-Y. Lee, K. Sunkavalli, and S. Joo Kim, "Fast video object segmentation by reference-guided mask propagation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7376–7385.

[53] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters–improve semantic segmentation by global convolutional network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4353–4361.

[54] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[55] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[56] 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 *Computer Vision and Pattern Recognition*, 2016.

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

[58] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[59] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 569–582, 2015.

[60] K. Xu, L. Wen, G. Li, L. Bo, and Q. Huang, "Spatiotemporal cnn for video object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1379–1388.

[61] J. Cheng, Y.-H. Tsai, W.-C. Hung, S. Wang, and M.-H. Yang, "Fast and accurate online video object segmentation via tracking parts," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7415–7424.

[62] Y. Chen, J. Pont-Tuset, A. Montes, and L. Van Gool, "Blazingly fast video object segmentation with pixel-wise metric learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1189–1198.

[63] J. Johnander, M. Danelljan, E. Brissman, F. S. Khan, and M. Felsberg, "A generative appearance model for end-to-end video object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8953–8962.

[64] L. Yang, Y. Wang, X. Xiong, J. Yang, and A. K. Katsaggelos, "Efficient video object segmentation via network modulation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6499–6507.

[65] Y.-T. Hu, J.-B. Huang, and A. G. Schwing, "Videomatch: Matching based video object segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 54–70.

[66] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5977–5986.

[67] Y.-H. Tsai, M.-H. Yang, and M. J. Black, "Video segmentation via object flow," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3899–3908.