A Unified Transformer Framework for Group-Based Segmentation: Co-Segmentation, Co-Saliency Detection and Video Salient Object Detection

Yukun Su, Jingliang Deng, Ruizhou Sun, Guosheng Lin, *Member, IEEE*, Hanjing Su, and Qingyao Wu, *Senior Member, IEEE*

***Abstract*—Humans tend to mine objects by learning from a group of images or several frames of video since we live in a dynamic world. In the computer vision area, many researchers focus on co-segmentation (CoS), co-saliency detection (CoSD) and video salient object detection (VSOD) to discover the co-occurrent objects. However, previous approaches design different networks for these similar tasks separately, and they are difficult to apply to each other. Besides, they fail to take full advantage of the cues among inter- and intra-feature within a group of images. In this paper, we introduce a unified framework to tackle these issues from a unified view, term as UFGS (Unified Framework for Group-based Segmentation). Specifically, we first introduce a transformer block, which views the image feature as a patch token and then captures their long-range dependencies through the self-attention mechanism. This can help the network to excavate the patch-structured similarities among the relevant objects. Furthermore, we propose an intra-MLP learning module to produce self-mask to enhance the network to avoid partial activation. Extensive experiments on four CoS benchmarks (PASCAL, iCoseg Internet and MSRC), three CoSD benchmarks (Cosal2015, CoSOD3k, and CocA) and five VSOD benchmarks (DAVIS$_{16}$, FBMS, ViSal, SegV2, and DAVSOD) show that our method outperforms other state-of-the-arts on three different tasks in both accuracy and speed by using the same network architecture, which can reach 140 FPS in real-time.**

***Index Terms*—Co-object, long-range dependency, transformer, activation.**

Manuscript received 25 July 2022; revised 29 January 2023 and 21 March 2023; accepted 28 March 2023. Date of publication 5 April 2023; date of current version 8 January 2024. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62272172, in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2023A1515012920, in part by Tip-Top Scientific and Technical Innovative Youth Talents of Guangdong Special Support Program under Grant 2019TQ05X200, in part by 2022 Tencent Wechat Rhino-Bird Focused Research Program (Tencent WeChat) under Grant RBFR2022008, in part by the Major Key Project of PCL under Grant PCL2021A09, in part by the National Research Foundation, Singapore under Its AI Singapore Programme (AISG) under Grant AISG-RP-2018-003, and in part by MOE AcRF Tier-1 Research under Grant RG95/20. The Associate Editor coordinating the review of this manuscript and approving it for publication was Prof. Lamberto Ballan. *(Corresponding authors: Guosheng Lin; Qingyao Wu.)*

Yukun Su, Jingliang Deng, and Ruizhou Sun are with the School of Software Engineering, Key Laboratory of Big Data and Intelligent Robot, Ministry of Education, South China University of Technology, Guangzhou 510006, China (e-mail: suyukun666@gmail.com; djl0628@126.com; ruizhousun@foxmail.com).

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

Hanjing Su is with the Tencent, Shenzhen 518000, China (e-mail: hanjingsu@gmail.com).

Qingyao Wu is with the School of Software Engineering, Pazhou Lab, South China University, Guangzhou 510006, China, and also with the Peng Cheng Laboratory, Shenzhen 518066, China (e-mail: qyw@scut.edu.cn).

Code is available at https://github.com/suyukun666/UFO.

This article has supplementary downloadable material available at https://doi.org/10.1109/TMM.2023.3264883, provided by the authors.

Digital Object Identifier 10.1109/TMM.2023.3264883

I. Introduction

OBJECT segmentation [1], [2], [3] and detection [4], [5] are the core tasks in computer vision. In our real world, the continuous emergence of massive group-based data and dynamic multi-frame data make deep learning more in the direction of human vision. As a result, more and more studies focus on co-segmentation (CoS) [6], [7], co-saliency detection (CoSD) [8], [9] and video salient object detection (VSOD) [10], [11]. Among them, these tasks all share common objects with the same attributes given a group of relevant images or within several adjacent frames in the video. They all essentially aim to discover and segment the co-occurrent object by imitating the human vision system.

However, as shown in Fig. 1 top, previous methods tend to design different networks on these tasks in isolation, which may hinder their application in real-world scenarios. Besides, the transferability and applicability of these methods are relatively poor. For example, some of the CoS [6], [12], CoSD [8], [13] and VSOD methods [10], [11] are well-trained on the same dataset [14], [15], they fail to achieve comparable performance on all benchmarks but only their specific task benchmarks. This illustrates that there may exist some limitations in previous approaches. To be specific, most of the co-object segmentation and saliency detection networks are based on matching strategies [16], [17], which enable the network to extract and propagate the feature representation not only to express the images' individual properties but also reflect the relevance and interaction among group-based images. Some recent works [6], [18] utilize spectral clustering [19] to mine the corresponding object regions. However, such methods are unstable, and they are deeply dependent on the upper-stream extracted features. Some researches [8], [20] adopt distance measure metric [21], [22] to model the relationship among image pixels, but they can only handle the pair-wise information and it is cumbersome to address group-wise relationships. Some others like [9], [13] try to use the CNN-based attention technique to establish the relationships among group-based images. However, convolution

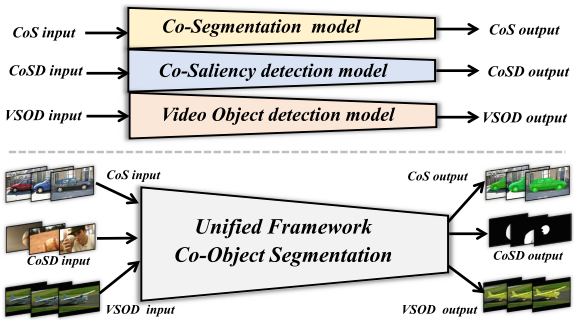

Fig. 1. Main purpose of our proposed framework. The co-object images can be fed into our network to yield the accurate masks in real-time using the same architecture.

operations produce local receptive fields and experience difficulty in capturing long-range feature dependencies among pixels, which may affect the learning representation of co-object regions. Furthermore, some of the VSOD approaches [11], [23] capture the pair-wise feature between frames with the help of optical flow [24], which greatly increases the network running cost and reduces the convenience of using the networks. Although some similar efforts like [25], [26] have exploited a unified framework for group-based image co-object segmentation, the former is designed based on traditional techniques and lacks advanced network architecture while the latter utilizes graph structure, which ignores the attention mechanism and fails to achieve competitive results on all the tasks.

To this end, we design a unified framework as shown in Fig. 1 bottom, term as **UFGS** (**U**nified **F**ramework for **G**roup-based **S**egmentation), to jointly address the aforementioned drawbacks. Specifically, to better reflect the relationships among group-based images, we first introduce a transformer block to insert into our network. It splits the image feature as a patch token and then captures their global dependency thanks to the self-attention mechanism and Multilayer Perceptron (MLP) structure. This can help the network learn complex spatial features and reflect long-range semantic correlations to excavate the patch-structured similarities among the relevant objects. The inherent global feature interaction capability of the visual transformer [27] frees us from the computationally expensive similarity matrices as in some previous methods [6]. Therefore, our method can achieve real-time performance. In addition to improving inter-collaboration in group-based images, we also propose an intra-MLP learning module to enhance the single image. As it is common that the encoder of the network only focuses on the most discriminative part of the objects [28], in order to avoid partial activation, we add the intra-MLP operation to produce global receptive fields. For each query pixel, it will match with its top-K potentially corresponding pixels, which can help the network learn divergently. Then we produce the self-masks and add them to the decoder to enhance the network.

Extensive experiments on four CoS benchmarks (i.e., PASCAL, iCoseg, Internet and MSRC), three CoSD benchmarks (i.e., Cosal2015, CoSOD3k, and CocA) and five VSOD benchmarks (i.e., DAVIS_{16}, FBMS, ViSal, SegV2 and DAVSOD) demonstrate the superiority of our approach and it can outperform the state-of-the-arts in both accuracy and speed (reach 140 FPS in real time) by using the same network architecture. The main contributions of our paper are the following:

- We propose a unified framework for group-based image co-object segmentation (**UFGS**). To the best of our knowledge, we take the early attempt to complete three different tasks (co-segmentation, co-saliency detection, and video salient object detection) using the same network architecture without using additional prior.
- We introduce the transformer block to capture the feature long-range dependencies among group-based images through the self-attention mechanism. Besides, we design an intra-MLP learning module to avoid partial activation to further enhance the network.

II. Related Work

Co-Segmentation (CoS): Co-Segmentation is introduced by Rother et al. [29], which aims to segment the common objects in pair images. Early conventional works like Gabor filters [30] and SIFT [31] tried to extract low-level image features and then detect image foreground. As deep learning recently emerges and demonstrates the success in many computer vision applications, more and more recent studies adopt deep visual features to train object co-segmentation. Chen et al. [32] and Li et al. [33] first proposed the siamese fully convolutional network to solve the object co-segmentation task with a mutual correlation layer. However, both of them can not achieve satisfactory performance and they can only deal with pair-wise images. Later, Li et al. [12] and Zhang [7] et al. both proposed group-wise networks using LSTM [34]. Although they can improve performance, they are computationally expensive because of the serial structure of the recurrent neural network. Besides, training such methods will have a risk, such as forgetting historical information. More recently, Chen et al. [35] proposed a matching strategy to jointly complete semantic matching and object co-segmentation. Such a method is targeted for bipartite matching, which is hard to apply to group-based segmentation. Zhang [6] later designed a spatial-semantic network with sub-cluster optimization. The cluster results are deeply dependent on the upper extracted features, and thus, it may make the training unstable. AGNN [26], [36] utilized a graph network, which establishes a fully connected graph. The nodes of the graph are composed of multi-frame images, and the edges of the graph are composed of the relationship between any two images. However, the conventional graph convolution networks will cause an over-smoothing problem when layers become deeper. Besides, the high-level image semantic latent may miss some useful cues due to the CNN downsampling operations.

Co-Saliency Detection (CoSD): Co-Saliency Detection is similar to Co-Segmentation. The main difference between them lies in that salient detection mimics the human vision system to distinguish the most visually distinctive regions [8], [37]. Previous standard methods [38], [39], [40] tried to use hand-crafted cues or super-pixels prior to discover the co-saliency from the images. Fu et al. [25] first designed a cluster-based algorithm for video segmentation and CoSD tasks, where the global correspondence between the multiple images is implicitly learned

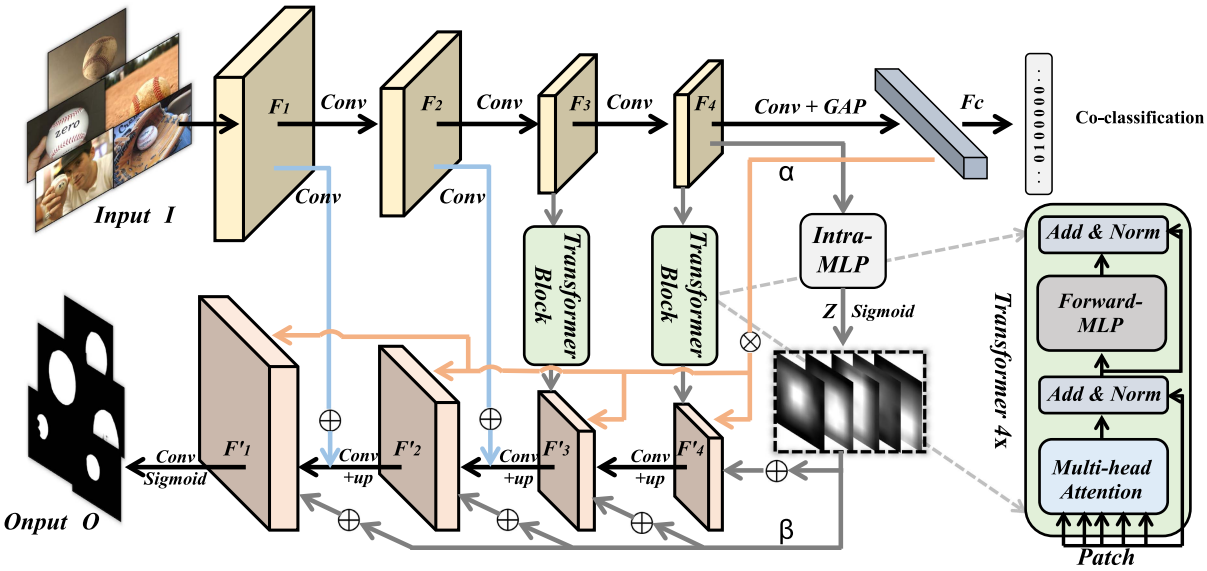

Fig. 2. Pipeline of our proposed method. The given group-based input images $\mathcal{I}$ are first fed into the encoder, yielding the multi-scale feature maps F. Then we employ transformer blocks (see Fig. 3 for more details) on the last two layers to capture the images' long-range correlations to model the patch-structured similarities among the relevant objects, which will output the updated co-attention layers F'. In addition, the last layer feature is passed through a classification sub-network and an intra-MLP module. The former exploits the co-category associated information and the latter produces the self-masks. Finally, they are combined with the updated co-attention layers to enhance the co-object semantic-aware regions in the pyramid network structure-like decoder to produce the output $\mathcal{O}$.

during the clustering process. Later, researchers pay more attention to exploring the deep-based models in a data-driven manner in various ways, i.e., co-category semantic fusion [6], [41], gradient-induced [37] and CNN-based self-attention [9], [13], *etc*. However, these works do not fully consider the global correlation among the group-based images due to the local convolution receptive fields. Some recent works [18], [42] exploited GCN to solve the non-local problem in CNN. However, in these methods, a large number of similarity matrices and adjacency matrices need to be constructed for the graph, which will slow down the networks and are computational-costly. Zhang et al. [8] proposed to use GW distance [43] to build the dense correlation volumes for image pixels. However, it has to select a target image and source images, which ignores the attention to the target image itself. Moreover, the distance metric problem in the network needs a sub-solver to optimize. This will cost more time to match. By contrast, our transformer block can capture the relationships of both inter-pixel within group-based images globally and intra-pixel within a single image locally.

Video Salient Object Detection (VSOD): In video salient object detection, since the content of each frame is highly correlated, it can be considered whose purpose is to capture long-range feature information among the adjacency frame. Traditional methods [44], [45], [46], [47] are usually based on classic heuristics in image salient object detection area. Subsequently, some works [48], [49] relied on 3D-convolution to capture the temporal information. However, the 3D convolution operation is very time-costly. In recent, more and more schemes [11], [46], [50] proposed to combine optical flow [24] to locate the representative salient objects in video frames. In practical usage, obtaining additional prior information such as the optical flow will make the deep learning network inconvenient, which can not be a real sense of an end-to-end network. More recently, some approaches exploited attention-based mechanisms to better establish the pair-wise relation in the area in consecutive frames. Lu et al. [51] designed a co-attention siamese network, but it can merely deal with the pair-wise frame input. Fan et al. [52] and Gu et al. [10] proposed a visual-attention-consistent module and a pyramid-constrained self-attention block to better capture the temporal dynamics cues, respectively. However, these models can not transfer well to the above tasks. Therefore, in this paper, we propose a unified framework to solve these problems in a more comprehensive way.

III. Methodology

A. Architecture Overview

Given a group of N images $\mathcal{I} = \{I^n\}_{n=1}^N$ that contain co-occurrent objects, our target is to produce the accurate mask outputs $\mathcal{O} = \{O^n\}_{n=1}^N$ that represent the share foregrounds. To achieve this goal, we propose a unified framework for group-based image co-object segmentation (**UFGS**). As depicted in Fig. 2, the overall architecture of our network is based on an encoder (i.e., VGG16 [53]) to extract multi-scale layer features $\{F_1, F_2, F_3, F_4\}$. The transformer blocks in the last two layers are responsible for matching the co-objects similarities in multi-resolution feature maps, producing the enhanced co-attention maps $\{F_3', F_4'\}$. Besides, the intra-MLP learning representation and the co-category semantic guidance are leveraged to combine with the updated maps in the decoder to enhance co-object regions. Specifically, the co-category embedding response α is multiplied by the decoder features, and the intra self-masks β are added to the decoder features. The encoder-decoder structure in our network is similar to the feature pyramid network [54], whose top-level features are fused by upsampling with low-level

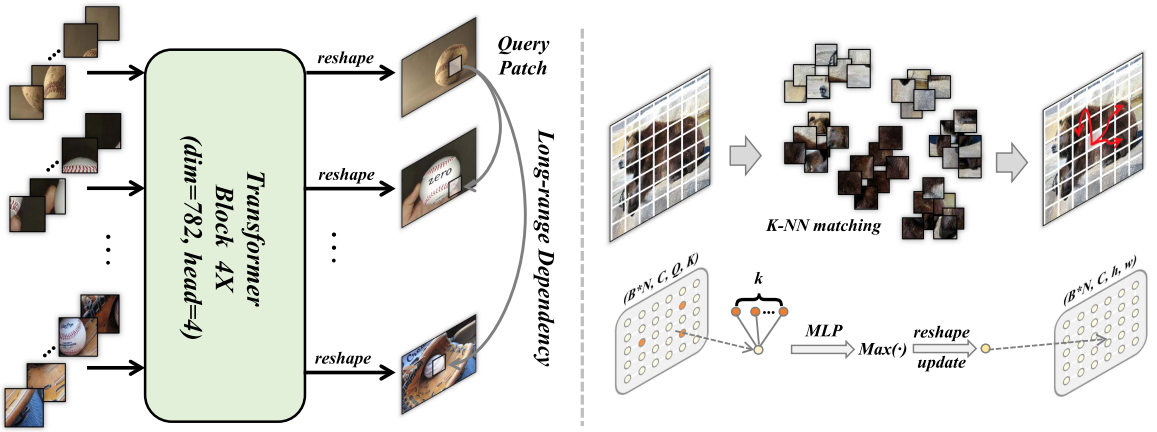

Fig. 3. Detailed illustration of the transformer block and the intra-MLP learning module. Left: A group of images $\mathcal{I}$ are reshaped into patches. The transformer encoder will output the enhanced patch embeddings implemented by the multi-head MLP. We then reshape all the patches into image-like feature maps. Right: For each query image feature patch, it will match with its top-K potentially corresponding patches. Then, it will be updated by aggregating different sub-region representations using MLP operation.

features in skip connection. Furthermore, the input of our network can be not only a group of images with relevant objects but also a multi-frame video.

B. Transformer Block

Preliminaries: In visual transformer [27], in order to handle 2D images, an input image will be divided into a sequence of flattened 2D patches. The transformer uses a constant latent vector size D in all its layers for each token embedding. Each layer consists of a Multi-Head Self-Attention (MHSA) layer and a Multilayer Perceptron (MLP) operation. Compared to other group-based image matching strategies and feature extraction methods, we consider that transformer block has two main advantages: (1) Although transformer is not specially designed for image matching, it has the intrinsic ability to capture global cues across all input patch tokens, which we have mentioned in Section II. (2) The transformer parallelizes the patch tokens, which is faster than some of the serial processing methods and can achieve real-time performance.

Semantic Patch Collaboration: The detailed workflow of the transformer block is shown in Fig. 3 left. Since we employ transformer blocks both on F_3 and F_4 layers in multi-resolution feature maps, for simplicity, we omit F_3 in the following. Concretely, we first reshape $F_4 \in \mathbb{R}^{B\times N\times C\times h\times w}$ (w and h denote the spatial size of the feature map, C is the feature dimension, B and N denote the batch size and group size, respectively) into a sequence of flattened patch tokens $T \in \mathbb{R}^{B\times C\times P}$, where $P = N * h * w$. All these tokens are then fed into the transformer blocks and yield the updated patch embeddings T' by:

$$\begin{aligned} T &= \mathrm{MHSA}(LN(T)) + T, \\ T' &= \mathrm{MLP}(LN(T)) + T, \end{aligned} \tag{1}$$

where the trainable linear projection maps the patches from the original C dimensions to D dimensions and back to C dimensions followed by the LayerNorm (LN) [55] function. Afterwards, we reshape T' back into an image-like feature map $F'_4 \in \mathbb{R}^{B\times N\times C\times h\times w}$. Due to the inductive bias [56] of convolution, the features extracted by the encoder convolution layers are not sensitive to the global location of features but only care about the existence of decisive features. The proposed transformer block can complement convolution and enhance the representations of the global co-object regions.

C. Intra-MLP Module

In addition to the image inter-collaboration, we also propose an intra-MLP learning module to activate more self-object areas within a single image. As shown in Fig. 3 right, the top layer feature map from the encoder is viewed as different patches. We consider that different patches will not just match with their nearest ones as in CNN (local receptive fields) since the long-distance patch features may share some similar responses (i.e., color and texture). Motivated by this, we can fuse the non-local semantic information to improve the object learning representation. Concretely, the top layer feature $F_4 \in \mathbb{R}^{B\times N\times C\times h\times w}$ is reshaped into $\overline{F}_4 \in \mathbb{R}^{B\times N\times C\times Q}$, where $Q = h * w$ is the number of patches. We then construct a matrix M that represents the similarity of each patch within a single image. Specifically, we use ℓ_2- distance to measure the relationship between the two arbitrary patches. Since we use normalized channel features, by removing the constant, the matrix $M \in \mathbb{R}^{B\times N\times Q\times Q}$ can be formulated as:

$$M = \overline{F}_4^T \overline{F}_4. \tag{2}$$

To avoid the patches match with themselves, the diagonal elements of the matrix are set to $-INF$. For each query patch, we perform KNN operation on the matrix to select its potentially corresponding target patches. Then, it will output a tensor in $Q \times K$ shape, which indicates the patches along with their top-K semantically related patches. After that, we can acquire the $\hat{F}_4 \in \mathbb{R}^{B\times N\times C\times Q\times K}$, and then we perform MLP with $MAX(\cdot)$ operations on it to get the updated feature $Z \in \mathbb{R}^{B\times N\times C\times Q}$ as:

$$Z = \mathrm{MAX}(\mathrm{MLP}(\hat{F}_4)), \tag{3}$$

where MAX is the element-wise maximum operation. This guarantees that the combination of MLP and symmetric function can arbitrarily approximate any continuous set function [57]. The purpose of this step is to combine the target feature with its top-K features appearance change information and learn through the perceptron. Finally, we reshape $Z \in \mathbb{R}^{B\times N\times C\times Q}$ back into $Z \in \mathbb{R}^{B\times N\times C\times h\times w}$ and employ $Sigmoid(\cdot)$ to Z to yield the self-masks β.

D. Network Training

Moreover, since the co-category information can also be used, the top layer F_4 is thus passed through a convolutional layer as [6], [41], following a GAP layer to yield a vector α. And the FC layer classifies the embedding α to predict the co-category labels $\hat{y}$. Then we combine both the α and self-masks β to enhance the decoder layer in a conditional normalization way [58] modulated with learned scale and bias. Specifically, the variables α are the learned scale modulation parameters that exclude the distractors in co-objects regions by using the rich semantic-aware clues. The variables β serve as the bias parameters complement and highlight the object targets by endowing spatial-aware information. The decoder leverages the skip connection structure to fuse the low-resolution layer features from the encoder. Ultimately, the network outputs the co-object masks.

Objective Function: Firstly, classification loss is used to update the gradient propagation for the semantic information as follow:

$$\mathcal{L}_{cls} = \mathcal{L}_{ce}(y, \hat{y}), \tag{4}$$

where $\mathcal{L}_{ce}$ is the cross-entropy loss, y is the ground-truth class label and $\hat{y}$ is the prediction. Besides, the Weighted Binary Cross-Entropy (WBCE) loss for pixel-wise segmentation is also adopted as follow:

$$\begin{aligned}\mathcal{L}_{wbce} = &-\frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W}\gamma G(i,j)log(P(i,j)) \\ &-(1-\gamma)(1-G(i,j))log(1-P(i,j)),\end{aligned} \tag{5}$$

where H and W denote the height and width of the image. $G(i,j) \in \{0,1\}$ is the ground truth mask and $P(i,j)$ is the predicted probability. γ is the ratio of all positive pixels over all pixels in images. Moreover, similar to [9], [59], IoU loss is also widely used to evaluate segmentation accuracy as follows:

$$\mathcal{L}_{iou} = 1 - \frac{\sum_{i=1}^{H}\sum_{j=1}^{W}P(i,j)G(i,j)}{\sum_{i=1}^{H}\sum_{j=1}^{W}[P(i,j)+G(i,j)-P(i,j)G(i,j)]}. \tag{6}$$

The whole framework is optimized by integrating all the aforementioned loss functions in an end-to-end manner:

$$\mathcal{L}_{total} = \mathcal{L}_{cls} + \mathcal{L}_{wbce} + \mathcal{L}_{iou}. \tag{7}$$

IV. Experiments

Datasets: Following [6], [7], we conduct experiments on four co-segmentation benchmarks including: PASCAL-VOC [60], iCoseg [61], Internet [31] and MSRC [62]. Among them, PASCAL-VOC is the most challenging dataset with 1,037 images of 20 categories. For co-saliency detection, our method is evaluated on the three largest and most challenging benchmark datasets, including Cosal2015 [63], CoCA [37], and CoSOD3k [64]. The CoSOD3k is the largest evaluation benchmark for real-world co-saliency proposed recently, which has a total of 160 categories with 3,316 images. And all of them contain multiple objects against a complex background. In terms of video salient object detection, we benchmark our method on five public datasets, i.e., DAVIS_{16} [65] (30 training videos and 20 validation videos), FBMS [66] (29 training videos and 30 testing videos), ViSal [67] (consists of 17 video sequences for testing), SegV2 [68] (consists of 13 clips for testing) and DAVSOD [52] (contains about 226 videos that cover diverse realistic-scenes, objects, instances and motions).

Evaluation Metrics: Two widely used measures, Precision ($\mathcal{P}$) and Jaccard index ($\mathcal{J}$), are used to evaluate the performance of object co-segmentation. For co-saliency detection and video salient object detection, we adopt four evaluation metrics for comparison, including the mean absolute error MAE [69], F-measure F_β [70], E-measure E_m [71], and S-measure S_m [72].

Training Details: The input image group $\mathcal{I}$ contains $N = 5$ images. The mini-batch size is set to $8 \times N$. For fair comparisons, we strictly follow the same settings as [9], [37] to use VGG16 [53] as the backbone and the images are all resized to 224×224 for training and testing unless otherwise stated. We use 4 transformer blocks with 4 multi-heads and K is set to 4 by default. And the feature dimension of the transformer linear projection function is 782. **(1):** On both co-segmentation (CoS) and co-saliency detection (CoSD) tasks, we follow [13], [41] to use the COCO-SEG [14] for training, which contains 200, 000 images belonging to 78 groups. And each image has a manually-labeled binary mask with co-category labels. We leverage the Adam algorithm [73] as the optimization strategy with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate is initially set to 1e-5 and reduced by a half every 25,000 steps. The whole training takes about 20 hours for total 100,000 steps. **(2):** On video salient detection task (VSOD), the common practice for most methods [10], [11], [74] are first pre-trained on the static saliency dataset, and then trained on video datasets. Following this scheme, we first load the weights pre-trained on the saliency task and combine DUT [15] dataset to train our network to avoid over-fitting. And then we froze the co-classification layer since we do not have class labels in VSOD task. Lastly, we train on the training set of DAVIS_{16} (30 clips) and FBMS (29 clips) as [11] and it takes about 4 hours. All the experiments are conducted on an RTX 3090 GPU. For all the runtime analysis, we report the results tested on Titan Xp GPU as in [10] for fair comparisons.

A. Ablation Studies

To explore each component of our proposed method, we conduct several ablation studies across three different tasks and select three representative datasets including PASCAL [60] from image-cosegmentaion, CoSOD3k [64] from co-saliency detection and DAVIS_{16} [65] from video salient detection to demonstrate their effectiveness.

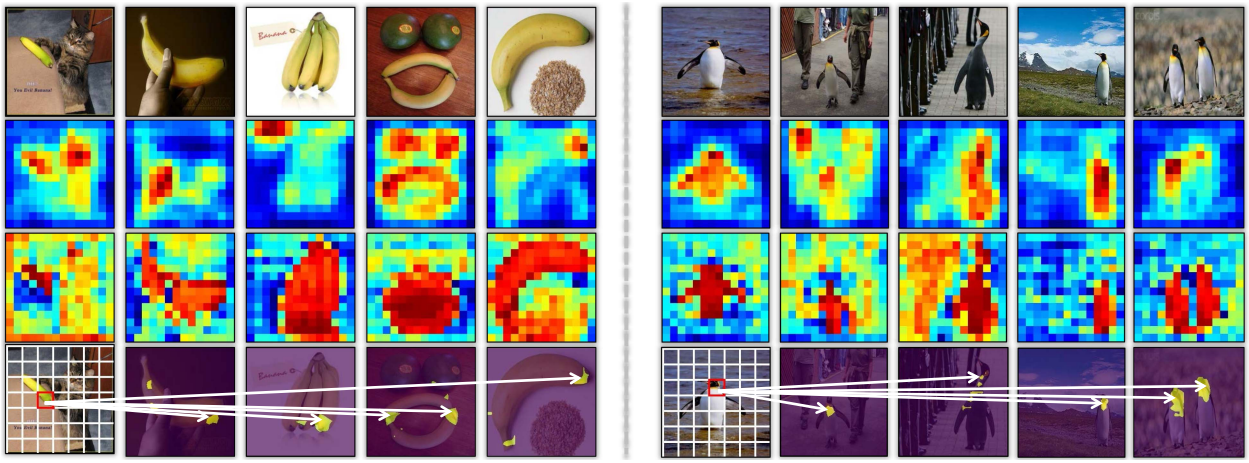

Fig. 4. Analysis of the transformer block. The 1st row: original group-based images input, note that we deliberately pick some hard examples with complex backgrounds and multi-objects. The 2nd row: the responded maps before transformer. The 3rd row: the responded maps of the patch tokens after transformer. The 4th row: the corresponding attention maps of the selected patches (marked with red rectangles in the query images).

TABLE I
ANALYSIS OF DIFFERENT MODULES AND THEIR COMBINATIONS

Baseline	α	β	Trans	PASCAL		CoSOD3K			DAVIS$_{16}$		
				$\mathcal{P}$	$\mathcal{J}$	*MAE*↓	S_m ↑	F_β ↑	*MAE*↓	S_m ↑	F_β ↑
✔				81.2	36.8	0.118	0.752	0.654	0.064	0.671	0.648
✔	✔			92.4	68.1	0.089	0.789	0.738	0.051	0.793	0.788
✔		✔		82.9	37.6	0.101	0.765	0.692	0.060	0.713	0.694
✔			✔	92.1	69.5	0.083	0.791	0.741	0.048	0.815	0.792
✔	✔	✔		92.5	68.6	0.082	0.788	0.741	0.045	0.826	0.805
✔	✔		✔	95.0	72.7	0.077	0.811	0.792	0.041	0.859	0.817
✔		✔	✔	92.8	69.6	0.084	0.805	0.781	0.039	0.855	0.811
✔	✔	✔	✔	**95.4**	**73.6**	**0.073**	**0.819**	**0.797**	**0.036**	**0.864**	**0.828**

TABLE II
ANALYSIS OF THE BLOCK AND MULTI-HEAD NUMBER IN TRANSFORMER

Transformer		PASCAL		CoSOD3K			DAVIS$_{16}$		
		$\mathcal{P}$	$\mathcal{J}$	*MAE*↓	S_m ↑	F_β ↑	*MAE*↓	S_m ↑	F_β ↑
Block = 2	Head = 2	94.0	71.8	0.081	0.805	0.783	0.044	0.854	0.817
	Head = 4	94.2	72.5	0.079	0.809	0.788	0.041	0.858	0.820
	Head = 6	94.6	73.0	0.076	0.812	0.791	0.039	0.860	0.819
Block = 4	Head = 2	94.8	72.9	0.075	0.814	0.796	0.038	0.860	0.824
	Head = 4	95.4	**73.6**	**0.073**	**0.819**	**0.797**	**0.036**	0.864	**0.828**
	Head = 6	95.0	73.5	**0.073**	0.816	0.793	0.038	0.859	0.822
Block = 6	Head = 2	95.1	72.9	**0.073**	0.817	0.795	**0.036**	0.861	0.825
	Head = 4	**95.6**	72.8	0.077	0.811	0.791	0.037	0.860	0.823
	Head = 6	95.0	73.1	0.076	0.812	0.792	0.040	**0.867**	0.826

TABLE III
ANALYSIS OF THE FEATURES FROM DIFFERENT STAGES IN THE ENCODER

Method	PASCAL		CoSOD3k			DAVIS$_{16}$		
	$\mathcal{P}$	$\mathcal{J}$	*MAE*↓	S_m ↑	F_β ↑	*MAE*↓	S_m ↑	F_β ↑
F_3 only	92.1	69.8	0.081	0.742	0.719	0.044	0.843	0.801
F_4 only	94.1	72.8	0.089	0.753	0.722	0.042	0.850	0.814
F_3 **and** F_4	**95.4**	**73.6**	**0.073**	**0.819**	**0.797**	**0.036**	**0.864**	**0.828**

What it learns? Firstly, we show what the transformer blocks learn (the output is from F_3') as in Fig. 4. As can be seen in the 3^{rd} row, the corresponding image patches of the co-object regions are highly activated after being operated by the transformer block compared to the 2^{nd} row, which illustrates that the attention mechanism in the transformer projection function can adaptively pay attention to the targeted areas and assign more weights to capture the global cues from all the images. In addition, we also visualize the query attention of the picked image patch (i.e., the endpoints of the "banana" and the "head" of the penguin) to show what it will match. The qualitative visualizations in the 3^{rd} row further validate that the transformer block can help the network model the long-range dependencies among different location pixels.

Comparisons to Baseline: In Table I, we investigate the effect of different proposed modules including co-category information α, intra-MLP masks β, transformer blocks and their combinations. As can be seen, each module can boost the performance of the baseline model to different degrees. Specifically, the proposed transformer block provides the main contributions to help improve the network performances. Whether used alone or in combination with the other two modules, it can outperform other alternatives by a large margin. This illustrates the effectiveness of each component in our framework.

Settings in Transformer: Table II shows the exploration of using different block and multi-head numbers in the transformer. We find that network performance can be improved when increasing the number of blocks and the multi-head attention mechanisms. However, keep increasing both the block and head numbers does not always bring gains to the network (i.e., block $= 6$), which will also cost much computational resources. We conjecture that large block and multi-head numbers will bring more parameters and redundant information for network learning. Therefore, we set both block and multi-head numbers to four in our paper.

Multi-Scale Features: In our method, we adopt transformer blocks on different top-level layers (i.e., F_3 and F_4). Since the low-level (i.e., F_1 and F_2) feature maps are large, which will greatly increase the patch number and is computationally expensive. Thus, we do not consider using them. Table III reveals that using both the features from the last two layers is better than using the features alone. This indicates that different level features can provide some vital **coarse-to-fine** information in the co-object segmentation task.

Self-Mask Production: In Table IV, we compare our intra-MLP learning module to standard convolution operation and

TABLE IV
ANALYSIS OF DIFFERENT OPERATIONS FOR SELF-MASKS PRODUCTION

Method	PASCAL		CoSOD3k			DAVIS$_{16}$		
	$\mathcal{P}$	$\mathcal{J}$	$MAE\downarrow$	$S_m\uparrow$	$F_\beta\uparrow$	$MAE\downarrow$	$S_m\uparrow$	$F_\beta\uparrow$
Conv	94.8	72.3	0.079	0.811	0.789	0.041	0.857	0.816
Non-Local [75]	95.1	72.9	0.078	0.813	0.792	**0.033**	0.860	0.821
Intra-MLP	**95.4**	**73.6**	**0.073**	**0.819**	**0.797**	0.036	**0.864**	**0.828**

TABLE V
ANALYSIS OF DIFFERENT METHODS' MATCHING TIME COST

Method	Cluster [6]	Graph [18]	GW-Distance [8]	**Transformer**
Time (*ms*)	49.1	50.2	>50	**5.4**

TABLE VI
ANALYSIS OF THE K NUMBER IN INTRA-MLP LEARNING MODULE

Method	PASCAL		CoSOD3k			DAVIS$_{16}$		
	$\mathcal{P}$	$\mathcal{J}$	$MAE\downarrow$	$S_m\uparrow$	$F_\beta\uparrow$	$MAE\downarrow$	$S_m\uparrow$	$F_\beta\uparrow$
$K=2$	94.5	72.8	0.079	0.811	0.789	0.042	**0.866**	0.823
$K=4$	**95.4**	**73.6**	**0.073**	**0.819**	0.797	**0.036**	0.864	**0.828**
$K=6$	95.2	73.3	0.076	0.812	**0.799**	0.040	0.858	0.821

TABLE VII
ANALYSIS OF THE DIM CHANNEL IN TRANSFORMER PROJECTION FUNCTION

Method	PASCAL		CoSOD3k			DAVIS$_{16}$		
	$\mathcal{P}$	$\mathcal{J}$	$MAE\downarrow$	$S_m\uparrow$	$F_\beta\uparrow$	$MAE\downarrow$	$S_m\uparrow$	$F_\beta\uparrow$
D = 512	94.7	72.3	0.081	0.813	0.790	**0.036**	0.860	0.823
D = 782	**95.4**	**73.6**	**0.073**	**0.819**	**0.797**	**0.036**	**0.864**	0.828
D = 1024	94.9	73.1	0.077	**0.819**	0.795	0.040	0.862	**0.830**

non-local [75]. We can observe that both our method and non-local can improve the performance by accepting more receptive fields to activate object regions compared to the standard convolution. Moreover, although non-local operation can improve the original convolution, it essentially uses some 1×1 convolutions to extract features and then reshape and multiply to get the output, which is still worse than our method.

Matching Time: We also exhibit the matching time of different methods in co-object regions mining in Table V. The results illustrate that our proposed method can not only better model co-object long-distance similarities, but also achieve the best performance in speed, which further validates the claim of the advantage of transformer block.

The number of K: We here explore the effect of the K number we use in the intra-MLP learning module. As shown in Table VI, we find that when $K = 4$, our network can achieve competitive performance. We conjecture that a small or too large K number may make the network learn less useful corresponding semantic information or redundant information that harms the network.

The Dim Channels: We further conduct an additional experiment to analyze the effect of the dim channel in our transformer block projection function. Since the top-level output features map channels are 512, and thus, we explore $\{D = 512, 782, 1024\}$, respectively. Table VII shows that when $D = 782$ the network performs the best. Note that keep enlarging the dim channel will not only bring no gain to the network, but also consume computing resources.

B. Comparison With State-of-the-Arts

Co-Segmentation: Table VIII presents the comparisons of our method with other existing CoS methods. Note that there is no standard specification for unifying the various methods on this task, and thus, we point out the respective backbone and input resolution in the table. By using the VGG16 [53] backbone, our framework outperforms all the other methods using the same backbone, even though some of them use a larger input size. Besides, compared to the stronger backbone methods like Chen et al. [35] and CycleSegNet [7], we can also achieve relatively comparable performance and even outperforms them. CARNN [12] adopts the VGG19 backbone and achieves the top precision on the Internet dataset. However, it is trained on the full COCO dataset. It contains 9 k images belonging to 118 groups, which is much larger than our training set. Furthermore, we also report the performance of some state-of-the-art methods in the co-saliency detection task (i.e., GCoNet [9] and CADC [13]). The results show that our method can also outperform them, which reflects that the transferability of such co-saliency detection methods is poor and they are unsuitable for co-segmentation. Besides, we also follow the setting in AGNN [26], [36] to retrain our network using the same VOC12 dataset for training. We can achieve **70.4** mean $\mathcal{J}$ against AGCNN (**60.8**) on PASCAL-VOC dataset [60], and achieve **81.3** mean $\mathcal{J}$ against AGCNN (**77.6**) on Internet [31] dataset. In general, our method achieves 5 best results and 2 s-best results on 4 CoS benchmarks, which is competitive. Fig. 5 shows some co-segmentation qualitative results.

Co-Saliency Detection: Likewise, Table IX presents the comparisons of our method with other state-of-the-arts in CoSD. Note that all methods use the VGG16 and the input size is 224 $\times$ 224 for fair comparisons unless otherwise specified. We can observe that our framework can also outperform other methods except for two second-best (i.e., $S_m = 0.860$ in Cosal2015 and $MAE = 0.073$ in CoSOD3k) results, and all the rest results are ranked the first. It is worth mentioning that unlike some of the previous methods (i.e., GCAGC [18] and CADC [13]), we do not require additional data [15], [100] and data augmentation strategies like Jigsaw in GICD [37] and copy-and-paste in CACD [13] to train the network. Moreover, we visualize the results of some examples (i.e., *pumpkin* and *soap*) with complex backgrounds and some foreground distractors in Fig. 6 top. Our method can yield more accurate co-object masks compared to other approaches, which validates the robustness and effectiveness of our method. Fig. 6 bottom shows the PR and the ROC curves of the compared methods, and our curves are higher than the other methods on the three challenging benchmarks.

Video Salient Object Detection: We further evaluate our method on VSOD benchmarks. As is shown in Table X, our proposed framework can once again achieve the best results on FBMS, ViSal and SegV2 datasets by using the VGG16 backbone and small input resolution but without optical flow. This shows that our method can effectively extract the object spatial appearance information and long-range temporal dependencies, which can also reach 140 FPS for real-time inference. Of note, since the input size (224 $\times$ 224) and architecture are consistent with the VSOD task, the execution time of CoS and

TABLE VIII
COMPARISONS OF OUR METHOD WITH THE OTHER STATE-OF-THE-ARTS ON COS DATASETS

Method	Backbone	Size	Params (M)	PASCAL		iCoseg		Internet		MSRC	
				$\mathcal{P}$	$\mathcal{J}$	$\mathcal{P}$	$\mathcal{J}$	$\mathcal{P}$	$\mathcal{J}$	$\mathcal{P}$	$\mathcal{J}$
Jerripothula et al. [76]$_{\text{TMM'2016}}$	-	-	-	85.2	45.0	91.9	72.0	88.9	64.0	88.7	71.0
Jerripothula et al. [77]$_{\text{CVPR'2017}}$	-	-	-	80.1	40.0	-	-	-	-	-	-
Wang et al. [78]$_{\text{TIP'2017}}$	ResNet50	300 × 300	-	84.3	52.0	93.8	77.0	-	-	90.9	73.0
Hsu et al. [79]$_{\text{IJCAI'2018}}$	VGG16	384 × 384	42.2	91.0	60.0	96.5	84.0	92.3	69.8	-	-
Chen et al. [32]$_{\text{ACCV'2018}}$	VGG16	512 × 512	32.7	-	59.8	-	84.0	-	73.1	-	77.7
Li et al. [33]$_{\text{ACCV'2018}}$	VGG16	512 × 512	> 69.3	94.2	64.5	95.1	84.2	93.5	72.6	95.2	82.9
CARNN[12]$_{\text{ICCV'2019}}$	VGG19	224 × 224	22.5	94.1	63.0	97.9	89.0	97.1	84.0	-	-
SSNM[6]$_{\text{AAAI'2020}}$	VGG16	224 × 224	54.5	93.7	66.0	96.5	88.0	92.3	67.0	94.3	76.3
Chen et al. [35]$_{\text{TPAMI'2020}}$	ResNet101	240 × 240	> 36.2	93.9	61.0	-	-	93.5	70.0	-	-
CycleSegNet[7]$_{\text{TIP'2021}}$	ResNet34	512 × 512	-	96.8	73.6	-	92.1	-	86.2	97.6	89.6
GCoNet†[9]$_{\text{CVPR'2021}}$	VGG16	224 × 224	28.0	93.5	69.2	96.9	89.7	92.5	70.5	94.0	80.8
CADC†[13]$_{\text{ICCV'2021}}$	VGG16	256 × 256	39.3	94.1	71.8	97.1	90.2	92.8	71.9	94.4	81.6
UFGS *(Ours)*	VGG16	224 × 224	45.4	**95.4**	**73.6**	**97.6**	**90.9**	**93.3**	**73.7**	**95.8**	**83.2**
UFGS *(Ours)*	VGG16	256 × 256	45.4	**96.9**	**75.7**	**98.1**	**92.3**	**95.2**	**74.6**	**97.8**	**84.3**

† Denotes the results using the publicly released code to re-complete. The best two results on each dataset are shown in red and blue.

Fig. 5. Qualitative results of our proposed network for "Bus", "Ballon" and "Bottle" objects.

TABLE IX
COMPARISONS OF OUR METHOD WITH THE OTHER STATE-OF-THE-ARTS ON COSD DATASETS

Method	Type	Params (M)	Cosal2015				CoSOD3k				CoCA			
			MAE↓	S_m ↑	E_m ↑	F_β ↑	*MAE*↓	S_m ↑	E_m ↑	F_β ↑	*MAE*↓	S_m ↑	E_m ↑	F_β ↑
BASNet[59]$_{\text{CVPR'2019}}$	Sin	87.1	0.097	0.820	0.846	0.784	0.122	0.753	0.791	0.696	0.195	0.589	0.623	0.397
PoolNet[80]$_{\text{CVPR'2019}}$	Sin	68.3	0.094	0.820	0.851	0.785	0.120	0.763	0.797	0.704	0.179	0.599	0.631	0.401
EGNet[81]$_{\text{ICCV'2019}}$	Sin	111.7	0.099	0.818	0.842	0.782	0.119	0.762	0.796	0.703	0.179	0.594	0.637	0.389
SCRN[82]$_{\text{ICCV'2019}}$	Sin	25.2	0.097	0.814	0.854	0.789	0.118	0.773	0.806	0.717	0.166	0.610	0.658	0.416
RCAN[83]$_{\text{IJCAI'2019}}$	Co	-	0.126	0.779	0.842	0.764	0.130	0.744	0.808	0.688	0.160	0.616	0.702	0.422
CSMG[84]$_{\text{CVPR'2019}}$	Co	-	0.130	0.774	0.818	0.777	0.157	0.711	0.723	0.645	0.124	0.632	0.734	0.503
GICD[37]$_{\text{ECCV'2020}}$	Co	278.0	0.071	0.842	0.884	0.834	0.089	0.778	0.831	0.743	0.125	0.658	0.701	0.513
SSNM[6]$_{\text{AAAI'2020}}$	Co	54.5	0.102	0.788	0.843	0.794	0.120	0.726	0.756	0.675	0.116	0.628	0.741	0.482
GCAGC[18]$_{\text{CVPR'2020}}$	Co	73.5	0.085	0.817	0.866	0.813	0.100	0.785	0.816	0.740	0.111	0.669	0.754	0.523
CoEGNet[64]$_{\text{TPAMI'2021}}$	Co	68.0	0.077	0.836	0.882	0.832	0.092	0.762	0.825	0.736	0.106	0.612	0.717	0.493
DeepACG[8]$_{\text{CVPR'2021}}$	Co	-	0.064	0.854	0.892	0.842	0.089	0.792	0.838	0.756	0.102	0.688	0.771	0.552
GCoNet[9]$_{\text{CVPR'2021}}$	Co	28.0	0.068	0.845	0.887	0.847	0.071	0.802	0.860	0.777	0.105	0.673	0.760	0.544
CADC†[13]$_{\text{ICCV'2021}}$	Co	39.3	0.064	0.866	0.906	0.862	0.096	0.801	0.840	0.759	0.132	0.681	0.744	0.548
GCAGC-CSD[85]$_{\text{TMM'2021}}$	Co	73.5	0.089	0.823	0.890	0.831	0.902	0.759	0.823	0.730	-	-	-	-
GWSCoSal[86]$_{\text{TMM'2022}}$	Co	-	0.068	0.846	0.886	-	-	-	-	-	-	-	-	-
UFGS *(Ours)*	Co	45.4	**0.064**	**0.860**	**0.906**	**0.865**	**0.073**	**0.819**	**0.874**	**0.797**	**0.095**	**0.697**	**0.782**	**0.571**

"Sin" and "Co" denote single and co-object image saliency object detection methods, respectively. † Denotes the results using a larger input size (256 × 256) and copy-and-paste augmentation strategy. The best two results on each dataset are shown in red and blue.

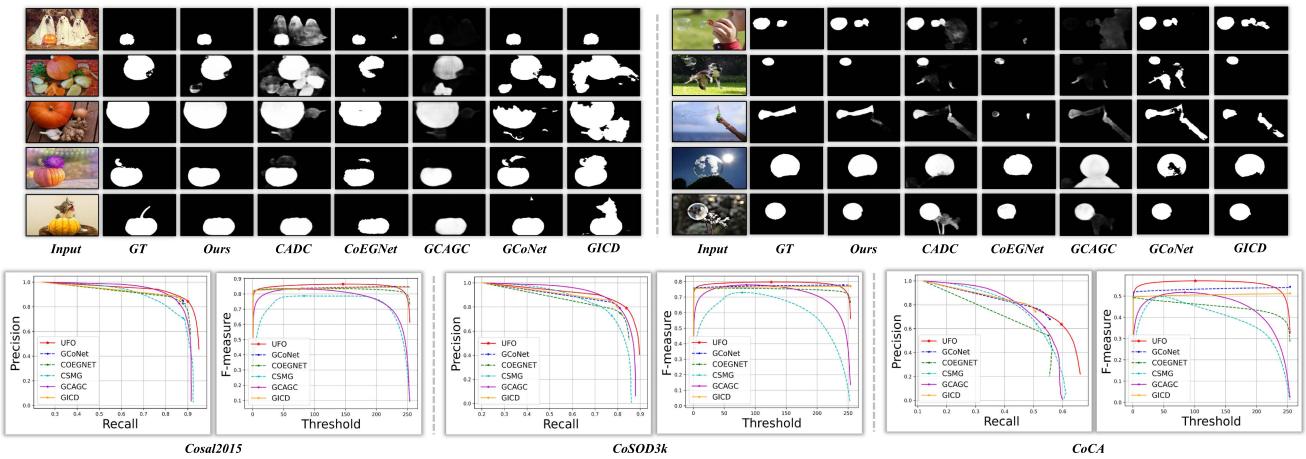

Fig. 6. Upper: Qualitative results of our method compared with other state-of-the-art methods. Bottom: The PR and F-measure curves on three benchmark datasets.

TABLE X
COMPARISONS OF OUR METHOD WITH THE OTHER STATE-OF-THE-ARTS ON VSOD DATASETS

Method	Setting	*Params (M)*	*OF*	*Sup*	*RT*	$DAVIS_{16}$			FBMS			ViSal			SegV2		
						MAE↓	S_m ↑	F_β ↑	*MAE*↓	S_m ↑	F_β ↑	*MAE*↓	S_m ↑	F_β ↑	*MAE*↓	S_m ↑	F_β ↑
SCOM[87]$_{TIP'2018}$	DCL[88]	-	✔	U	38.8	0.048	0.832	0.783	0.079	0.794	0.797	0.122	0.762	0.831	0.030	0.815	0.764
MBNM[89]$_{ECCV'2018}$	DeepLab	-	✔	U	2.6	0.031	0.887	0.861	0.047	0.857	0.816	0.020	0.898	0.883	0.026	0.809	0.716
PDBM[90]$_{ECCV'2018}$	Res50-473	59.2	✗	U	0.05	0.028	0.882	0.855	0.064	0.851	0.821	0.032	0.907	0.888	0.024	0.864	0.800
SRP[91]$_{TIP'2019}$	-	-	✔	U	17.0	0.070	0.662	0.660	0.134	0.648	0.671	0.092	-	0.752	0.095	-	0.683
MESO[23]$_{TMM'2019}$	-	-	✔	U	50.3	0.070	0.718	0.660	0.134	0.635	0.618	-	-	-	-	-	-
LTSI[92]$_{TIP'2019}$	VGG16-500	-	✔	U	1.4	0.034	0.876	0.850	0.087	0.805	0.799	0.027	0.922	0.909	0.028	0.827	0.862
RSE[46]$_{TCSVT'2019}$	-	-	✔	U	48.2	0.063	0.748	0.698	0.128	0.670	0.652	-	-	-	-	-	-
SSAV[52]$_{CVPR'2019}$	Res50-473	-	✗	U	0.05	0.028	0.893	0.861	0.040	0.879	0.865	0.020	0.943	0.939	0.023	0.851	0.801
RCR[50]$_{ICCV'2019}$	Res50-448	56.4	✔	S	0.04	0.027	0.886	0.848	0.053	0.872	0.859	-	-	-	-	-	-
CAS[93]$_{TNNLS'2020}$	Res50	-	✗	U	-	0.032	0.873	0.860	0.056	0.856	0.863	-	-	-	0.029	0.820	0.847
PCSA[10]$_{AAAI'2020}$	MobV3-448	46.6	✗	U	0.009	0.022	0.902	0.880	0.040	0.868	0.837	0.017	0.946	0.940	0.025	0.865	0.810
DFNet[74]$_{ECCV'2020}$	DeepLab	64.7	✗	U	-	0.018	-	0.899	0.054	-	0.833	0.017	-	0.927	-	-	-
FSNet[11]$_{ICCV'2021}$	Res101-352	$\gg 2.3$	✔	U	0.08	0.020	0.920	0.907	0.041	0.890	0.888	-	-	-	0.023	0.870	0.772
ReuseVOS†[94]$_{CVPR'2021}$	Res18-480	30.5	✗	S	0.02	0.019	0.883	0.865	0.027	0.888	0.884	0.020	0.928	0.933	0.025	0.844	0.832
SFEN[95]$_{TMM'2021}$	Res18-384	-	✔	U	0.019	0.019	0.923	0.906	-	-	-	0.017	0.946	0.938	-	-	-
UFGS *(Ours)*	VGG16-224	45.4	✗	U	**0.007**	0.036	0.864	0.828	**0.028**	**0.894**	**0.890**	**0.011**	**0.953**	**0.940**	**0.022**	**0.892**	**0.863**
UFGS *(Ours)*	VGG16-224	> 45.4	✔	U	0.01	**0.015**	0.918	**0.906**	**0.031**	**0.891**	**0.888**	**0.013**	**0.959**	**0.951**	**0.013**	**0.899**	**0.869**
UFGS *(Ours)*	HRNet-224	48.6	✗	U	0.01	0.028	0.881	0.842	0.026	0.899	0.893	0.012	0.958	0.948	0.018	0.897	0.866
UFGS *(Ours)*	HRNet-224	> 48.6	✔	U	0.03	0.013	0.921	0.907	0.033	0.888	0.887	0.011	0.962	0.956	0.012	0.901	0.867

Some of the results are borrowed from [11]. *OF* denotes the optical flow. *RT* denotes the runtime (s). "U": Unsupervised method. "S" :Semi-supervised method. "Res" denotes resnet [96], "Mob" denotes moiblenet [97], and the number behind them is the input resolution. † denotes video segmentation methods trained on DAVIS17 [98] and youtube-VOS [99] datasets, whose results are acquired from their released code weights. The best two results on each dataset are shown in red and blue except for adopting our hrnet backbone.

CoSD tasks is the same as VSOD task. Moreover, the proposed framework is an *unsupervised* VSOD method, which can outperform some *semi-supervised* methods [50], [94] without any bells and whistles. The reason why we fail to achieve the best results on the DAVIS is that in some cases the background of co-occurring objects is incorrectly segmented (*e.g.*, foreground: the dancing girl; background: the audience. They all belong to the object of "people"). To alleviate this issue, we combine the flow cues to make our network pay more attention to the foreground moving salient objects and using optical flow in our network is straightforward. Specifically, the images and their corresponding optical flow are passed through the share-encoder at the same time, and the enhanced image features to be fed into subsequent networks can be obtained by multiplying the flow features. More details of the usage of optical flow can be referred to the supplementary material. The quantitative results show that we can achieve new satisfactory results. The performances on FBMS drop a little since the acquired optical flow information are poor. Note that using the optical flow is not our framework's main purpose and contribution, we recommend using the multi-frame images alone, which can already achieve state-of-the-art performance in most tasks. Besides, we adopt a stronger backbone HRNet [103] to conduct experiments. Some qualitative results on the four VSOD datasets are shown in Fig. 7. Furthermore, we conduct an additional experiment on a more challenging instance-level VSOD dataset [52]. As shown in Table XI, our proposed framework can achieve competitive results in terms of metric F_β. Compared to the previous works that are specifically designed for video salient object detection, our method can achieve satisfactory results, which is acceptable.

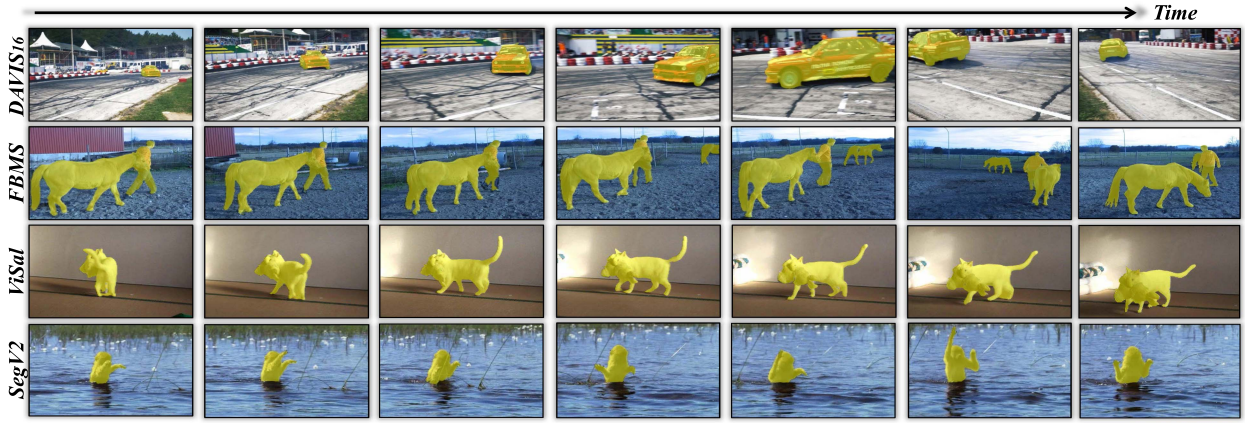

Fig. 7. Qualitative results of our proposed framework on DAVIS, FBMS, ViSal and SegV2 datasets, respectively. More visualizations can be referred to the supplementary material.

Fig. 8. Left: Visualizations of co-localization. The 1st row is the input images, the 2nd row is our predicted masks, and the 3rd row is the localization results. The ground-truth bounding box is in red, and the predicted bounding box is **green**. Right: Visualizations of image inpainting. For each row, the input and the perdition of the model are concatenated.

TABLE XI
COMPARISON WITH SOTA VIDEO SALIENT DETECTION METHODS ON DAVSOD [52] DATASET

Method		SSAV [52]	SVSN [101]	MGAN [102]	RCRN [50]	PCSA [10]	SFEN [95]	Ours
DAVSOD	$F_\beta \uparrow$	0.626	0.578	0.664	0.650	0.651	0.686	**0.692**
	$MAE \downarrow$	0.083	0.114	0.075	0.078	0.077	**0.073**	0.075
	$S_m \uparrow$	0.719	0.673	0.733	0.723	0.724	**0.769**	0.764

C. Failure Cases

As shown in Fig. 9 (Top: input image; Middle: ground-truth; Bottom: perdition), we present some failure cases of our framework. Specifically, since our method does not consider the edge information, and thus, some boundary of prediction is not so ideal, and some slender parts are missing. We leave this issue and consider addressing it in future work.

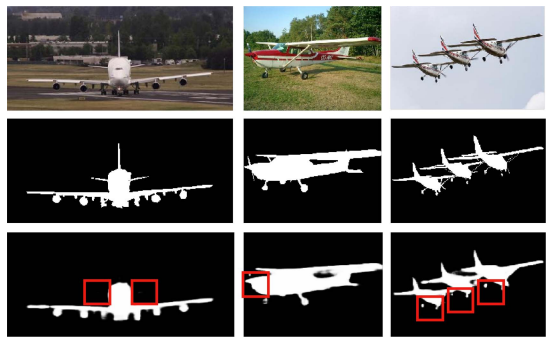

Fig. 9. Failure cases of our proposed framework.

D. More Applications

Co-Localization: We try to conduct an additional experiment on CUB-2011 [104] dataset for co-localization task, which also aims to simultaneously localize objects of the same class across a set of distinct images [105]. As shown in Fig. 8 left, we first generate the object masks on the co-objects. Then, we obtain the green predated box according to the binary masks. As can be seen, our predicted bounding boxes are close to the GT bounding boxes.

Video Inpainting: To further verify our framework's unity and effectiveness, we conduct a more difficult video inpainting task. Specifically, we simply modify the last sigmoid layer to the RGB layer, which yields the inpainted frames. And we follow the settings of the previous work [106] and train on the YouTube-VOS [99] training set. Surprisingly, our framework can achieve PSNR: 33.54 and SSIM: 0.9682 against the SOTA method [106] (PSNR: 33.16; SSIM: 0.9673). These results validate our network can well solve different image and video domain problems from a unified view. Fig. 8 right shows some video inpainting visualization examples.

Bullet-Chat Blocking: Bullet-chat blocking aims to prevent the salient objects in the video from being occluded by the viewer's barrage. It has to segment out the object and yield an accurate mask to block the bullet-chat text in the foreground. To

validate the generality of our method, we use a video sample from the public website.[1] And then we adopt the OpenCV [107] toolbox to generate some pseudo bullet-chat that constantly slides over the video. Finally, we can produce a new bullet-chat blocking video as we provide in the multimedia supplementary material. Such kinds of interesting and practical applications well reflect the scalability of our method in computer vision.

V. Conclusion

In this paper, we propose a unified framework for group-based image segmentation, which can conduct co-segmentation, co-saliency detection and video salient object detection tasks. The proposed transformer block and intra-MLP module can both help the network well capture the inter-collaborative and intra-activated information. The competitive results on 12 benchmarks validate the effectiveness of our method. We take an early attempt to unify the deep learning network for different tasks, and we hope this finding will encourage the development of more follow-up research that simplifies the network in other domains.

References

[1] 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 Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[2] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 801–818.

[3] Y. Su, R. Sun, G. Lin, and Q. Wu, "Context decoupling augmentation for weakly supervised semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 7004–7014.

[4] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[5] H. Law and J. Deng, "CornerNet: Detecting objects as paired keypoints," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 734–750.

[6] K. Zhang, J. Chen, B. Liu, and Q. Liu, "Deep object co-segmentation via spatial-semantic network modulation," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 12813–12820.

[7] C. Zhang, G. Li, G. Lin, Q. Wu, and R. Yao, "CycleSegNet: Object co-segmentation with cycle refinement and region correspondence," *IEEE Trans. Image Process.*, vol. 30, pp. 5652–5664, 2021.

[8] K. Zhang, M. Dong, B. Liu, X.-T. Yuan, and Q. Liu, "DeepACG: Co-saliency detection via semantic-aware contrast Gromov-Wasserstein distance," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 13703–13712.

[9] Q. Fan et al., "Group collaborative learning for co-salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 12288–12298.

[10] Y. Gu et al., "Pyramid constrained self-attention network for fast video salient object detection," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 10869–10876.

[11] G.-P. Ji et al., "Full-duplex strategy for video object segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 4922–4933.

[12] B. Li, Z. Sun, Q. Li, Y. Wu, and A. Hu, "Group-wise deep object co-segmentation with co-attention recurrent neural network," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 8519–8528.

[13] N. Zhang, J. Han, N. Liu, and L. Shao, "Summarize and search: Learning consensus-aware dynamic convolution for co-saliency detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 4167–4176.

[14] T.-Y. Linet al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2014, pp. 740–755.

[15] L. Wang et al., "Learning to detect salient objects with image-level supervision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 136–145.

[1][Online]. Available: https://www.youtube.com/watch?v=kpUYkG0FAYk&t=146~s

[16] J. Ma, X. Jiang, A. Fan, J. Jiang, and J. Yan, "Image matching from handcrafted to deep features: A survey," *Int. J. Comput. Vis.*, vol. 129, no. 1, pp. 23–79, 2021.

[17] Y. Liu, L. Zhu, M. Yamada, and Y. Yang, "Semantic correspondence as an optimal transport problem," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 4463–4472.

[18] K. Zhang et al., "Adaptive graph convolutional network with attention graph clustering for co-saliency detection," in *Proc. IEEE/CVF Conf. Comput. Vis. pattern Recognit.*, 2020, pp. 9050–9059.

[19] U. V. Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.

[20] C. Zhang, Y. Cai, G. Lin, and C. Shen, "DeepEMD: Few-shot image classification with differentiable earth mover's distance and structured classifiers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12203–12213.

[21] J. Solomon, G. Peyré, V. G. Kim, and S. Sra, "Entropic metric alignment for correspondence problems," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 1–13, 2016.

[22] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *Int. J. Comput. Vis.*, vol. 40, no. 2, pp. 99–121, 2000.

[23] M. Xu, B. Liu, P. Fu, J. Li, and Y. H. Hu, "Video saliency detection via graph clustering with motion energy and spatiotemporal objectness," *IEEE Trans. Multimedia*, vol. 21, no. 11, pp. 2790–2805, Nov. 2019.

[24] E. Ilg et al., "FlowNet 2.0: Evolution of optical flow estimation with deep networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2462–2470.

[25] H. Fu, X. Cao, and Z. Tu, "Cluster-based co-saliency detection," *IEEE Trans. Image Process.*, vol. 22, no. 10, pp. 3766–3778, Oct. 2013.

[26] W. Wang, X. Lu, J. Shen, D. J. Crandall, and L. Shao, "Zero-shot video object segmentation via attentive graph neural networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9236–9245.

[27] A. Dosovitskiyet al., "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[28] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2921–2929.

[29] C. Rother, T. Minka, A. Blake, and V. Kolmogorov, "Cosegmentation of image pairs by histogram matching-incorporating a global constraint into MRFs," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 993–1000.

[30] D. S. Hochbaum and V. Singh, "An efficient algorithm for co-segmentation," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, 2009, pp. 269–276.

[31] M. Rubinstein, A. Joulin, J. Kopf, and C. Liu, "Unsupervised joint object discovery and segmentation in internet images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 1939–1946.

[32] H. Chen, Y. Huang, and H. Nakayama, "Semantic aware attention based deep object co-segmentation," in *Proc. Asian Conf. Comput. Vis.*, Springer, 2018, pp. 435–450.

[33] W. Li, O. Hosseini Jafari, and C. Rother, "Deep object co-segmentation," in *Proc. Asian Conf. Comput. Vis.*, 2018, pp. 638–653.

[34] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[35] Y.-C. Chen, Y.-Y. Lin, M.-H. Yang, and J.-B. Huang, "Show, match and segment: Joint weakly supervised learning of semantic matching and object co-segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3632–3647, Oct. 2021.

[36] X. Lu, W. Wang, J. Shen, D. Crandall, and L. Van Gool, "Segmenting objects from relational visual data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 7885–7897, Nov. 2022.

[37] Z. Zhang, W. Jin, J. Xu, and M.-M. Cheng, "Gradient-induced co-saliency detection," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 455–472.

[38] H. Li, F. Meng, and K. N. Ngan, "Co-salient object detection from multiple images," *IEEE Trans. Multimedia*, vol. 15, no. 8, pp. 1896–1909, Dec. 2013.

[39] H. Song, Z. Liu, Y. Xie, L. Wu, and M. Huang, "RGBD co-saliency detection via bagging-based clustering," *IEEE Signal Process. Lett.*, vol. 23, no. 12, pp. 1722–1726, Dec. 2016.

[40] K. R. Jerripothula, J. Cai, and J. Yuan, "CATS: Co-saliency activated tracklet selection for video co-localization," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 187–202.

[41] C. Wang, Z.-J. Zha, D. Liu, and H. Xie, "Robust deep co-saliency detection with group semantic," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 8917–8924.

[42] B. Jiang, X. Jiang, A. Zhou, J. Tang, and B. Luo, "A unified multiple graph learning and convolutional network model for co-saliency estimation," in *Proc. 27th ACM Int. Conf. Multimedia*, 2019, pp. 1375–1382.

[43] F. Mémoli, "The Gromov–Wasserstein distance: A brief overview," *Axioms*, vol. 3, no. 3, pp. 335–341, 2014.

[44] Y. Wei, F. Wen, W. Zhu, and J. Sun, "Geodesic saliency using background priors," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 29–42.

[45] W. Wang, J. Shen, R. Yang, and F. Porikli, "Saliency-aware video object segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 20–33, Jan. 2018.

[46] M. Xu et al., "Video salient object detection via robust seeds extraction and multi-graphs manifold propagation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 7, pp. 2191–2206, Jul. 2020.

[47] X. Zhou, Z. Liu, C. Gong, and W. Liu, "Improving video saliency detection via localized estimation and spatiotemporal refinement," *IEEE Trans. Multimedia*, vol. 20, no. 11, pp. 2993–3007, Nov. 2018.

[48] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4489–4497.

[49] S. Mahadevan et al., "Making a case for 3D convolutions for object segmentation in videos," 2020, *arXiv:2008.11516*.

[50] P. Yan et al., "Semi-supervised video salient object detection using pseudo-labels," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 7284–7293.

[51] X. Lu et al., "See more, know more: Unsupervised video object segmentation with co-attention siamese networks," in *Proc. IEEE/CVF Conf. Comput. Vis. pattern Recognit.*, 2019, pp. 3623–3632.

[52] D.-P. Fan, W. Wang, M.-M. Cheng, and J. Shen, "Shifting more attention to video salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8554–8564.

[53] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[54] T.-Y. Lin et al., "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2117–2125.

[55] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.

[56] A. Goyal and Y. Bengio, "Inductive biases for deep learning of higher-level cognition," *Proc. Royal Soc. A*, vol. 478, no. 2266, 2020, Art. no. 20210068.

[57] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 652–660.

[58] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2337–2346.

[59] X. Qin et al., "BASNet: Boundary-aware salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7479–7489.

[60] M. Everingham et al., "The pascal visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, 2015.

[61] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen, "iCoseg: Interactive co-segmentation with intelligent scribble guidance," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3169–3176.

[62] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "TextonBoost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 1–15.

[63] D. Zhang, J. Han, C. Li, and J. Wang, "Co-saliency detection via looking deep and wide," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 2994–3002.

[64] D.-P. Fan et al., "Re-thinking co-salient object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 8, pp. 4339–4354, Aug. 2022.

[65] F. Perazzi et al., "A benchmark dataset and evaluation methodology for video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 724–732.

[66] P. Ochs, J. Malik, and T. Brox, "Segmentation of moving objects by long term video analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 6, pp. 1187–1200, Jun. 2014.

[67] W. Wang, J. Shen, and L. Shao, "Consistent video saliency using local gradient flow optimization and global refinement," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 4185–4196, Nov. 2015.

[68] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg, "Video segmentation by tracking many figure-ground segments," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 2192–2199.

[69] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung, "Saliency filters: Contrast based filtering for salient region detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 733–740.

[70] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1597–1604.

[71] D.-P. Fan et al., "Enhanced-alignment measure for binary foreground map evaluation," 2018, *arXiv:1805.10421*.

[72] D.-P. Fan, M.-M. Cheng, Y. Liu, T. Li, and A. Borji, "Structure-measure: A new way to evaluate foreground maps," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4548–4557.

[73] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[74] M. Zhen et al., "Learning discriminative feature with CRF for unsupervised video object segmentation," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2020, pp. 445–462.

[75] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7794–7803.

[76] K. R. Jerripothula, J. Cai, and J. Yuan, "Image co-segmentation via saliency co-fusion," *IEEE Trans. Multimedia*, vol. 18, no. 9, pp. 1896–1909, Sep. 2016.

[77] K. R. Jerripothula, J. Cai, J. Lu, and J. Yuan, "Object co-skeletonization with co-segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3881–3889.

[78] C. Wang, H. Zhang, L. Yang, X. Cao, and H. Xiong, "Multiple semantic matching on augmented n-partite graph for object co-segmentation," *IEEE Trans. Image Process.*, vol. 26, no. 12, pp. 5825–5839, Dec. 2017.

[79] K.-J. Hsu et al., "Co-attention CNNs for unsupervised object co-segmentation," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 748–756.

[80] J.-J. Liu, Q. Hou, M.-M. Cheng, J. Feng, and J. Jiang, "A simple pooling-based design for real-time salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3917–3926.

[81] J.-X. Zhao et al., "EGNet: Edge guidance network for salient object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 8779–8788.

[82] Z. Wu, L. Su, and Q. Huang, "Stacked cross refinement network for edge-aware salient object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 7264–7273.

[83] B. Li, Z. Sun, L. Tang, Y. Sun, and J. Shi, "Detecting robust co-saliency with recurrent co-attention neural network," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 818–825.

[84] K. Zhang, T. Li, B. Liu, and Q. Liu, "Co-saliency detection via mask-guided fully convolutional networks with multi-scale label smoothing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3095–3104.

[85] T. Li et al., "Image co-saliency detection and instance co-segmentation using attention graph clustering based graph convolutional network," *IEEE Trans. Multimedia*, vol. 24, pp. 492–505, 2021.

[86] X. Qian, Y. Zeng, W. Wang, and Q. Zhang, "Co-saliency detection guided by group weakly supervised learning," *IEEE Trans. Multimedia*, early access, Apr. 10, 2022, doi: 10.1109/TMM.2022.3167805.

[87] Y. Chen et al., "SCOM: Spatiotemporal constrained optimization for salient object detection," *IEEE Trans. Image Process.*, vol. 27, no. 7, pp. 3345–3357, Jul. 2018.

[88] G. Li and Y. Yu, "Deep contrast learning for salient object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 478–487.

[89] S. Li, B. Seybold, A. Vorobyov, X. Lei, and C.-C. J. Kuo, "Unsupervised video object segmentation with motion-based bilateral networks," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 207–223.

[90] H. Song, W. Wang, S. Zhao, J. Shen, and K.-M. Lam, "Pyramid dilated deeper convLSTM for video salient object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 715–731.

[91] R. Cong et al., "Video saliency detection via sparsity-based reconstruction and propagation," *IEEE Trans. Image Process.*, vol. 28, no. 10, pp. 4819–4831, Oct. 2019.

[92] C. Chen, G. Wang, C. Peng, X. Zhang, and H. Qin, "Improved robust video saliency detection based on long-term spatial-temporal information," *IEEE Trans. Image Process.*, vol. 29, pp. 1090–1100, 2019.

[93] Y. Ji, H. Zhang, Z. Jie, L. Ma, and Q. J. Wu, "CASNet: A cross-attention siamese network for video salient object detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 6, pp. 2676–2690, Jun. 2021.

[94] H. Park, J. Yoo, S. Jeong, G. Venkatesh, and N. Kwak, "Learning dynamic network using a reuse gate function in semi-supervised video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8405–8414.

[95] Y. Kong, Y. Wang, A. Li, and Q. Huang, "Self-sufficient feature enhancing networks for video salient object detection," *IEEE Trans. Multimedia*, vol. 25, pp. 557–571, 2021.

[96] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[97] A. Howard et al., "Searching for MobileNetv3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1314–1324.
[98] J. Pont-Tuset et al., "The 2017 DAVIS challenge on video object segmentation," 2017, *arXiv:1704.00675*.
[99] N. Xu et al., "YouTube-VOS: A large-scale video object segmentation benchmark," 2018, *arXiv:1809.03327*.
[100] T. Liu et al., "Learning to detect a salient object," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 2, pp. 353–367, Feb. 2011.
[101] Z. Wang, X. Yan, Y. Han, and M. Sun, "Ranking video salient object detection," in *Proc. 27th ACM Int. Conf. Multimedia*, 2019, pp. 873–881.
[102] H. Li, G. Chen, G. Li, and Y. Yu, "Motion guided attention for video salient object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 7274–7283.
[103] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5693–5703.
[104] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD birds-200-2011 dataset," Tech. Rep., 2010-001, California Inst. Technol., Pasadena, CA, USA, 2011.
[105] K. Tang, A. Joulin, L.-J. Li, and L. Fei-Fei, "Co-localization in real-world images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1464–1471.
[106] R. Liu et al., "FuseFormer: Fusing fine-grained information in transformers for video inpainting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 14040–14049.
[107] G. Bradski, "The opencv library," *Dr Dobb's J.: Softw. Tools Professional Programmer*, vol. 25, no. 11, pp. 120–123, 2000.