GAPNet: A Lightweight Framework for Image and Video Salient Object Detection via Granularity-Aware Paradigm

Yu-Huan Wu[1], Wei Liu[1†], Zi-Xuan Zhu[2], Zizhou Wang[1], Yong Liu[1] and Liangli Zhen[1†]

[1]Institute of High Performance Computing (IHPC), A*STAR, Singapore 138632.
[2]VCIP, College of Computer Science, Nankai University, Tianjin, China 300350.

[†]Corresponding authors

Abstract

Recent salient object detection (SOD) models predominantly rely on heavyweight backbones, incurring substantial computational cost and hindering their practical application in various real-world settings, particularly on edge devices. This paper presents GAPNet, a lightweight network built on the granularity-aware paradigm for both image and video SOD. We assign saliency maps of different granularities to supervise the multi-scale decoder side-outputs: coarse object locations for high-level outputs and fine-grained object boundaries for low-level outputs. Specifically, our decoder is built with granularity-aware connections which fuse high-level features of low granularity and low-level features of high granularity, respectively. To support these connections, we design granular pyramid convolution (GPC) and cross-scale attention (CSA) modules for efficient fusion of low-scale and high-scale features, respectively. On top of the encoder, a self-attention module is built to learn global information, enabling accurate object localization with negligible computational cost. Unlike traditional U-Net-based approaches, our proposed method optimizes feature utilization and semantic interpretation while applying appropriate supervision at each processing stage. Extensive experiments show that the proposed method achieves a new state-of-the-art performance among lightweight image and video SOD models. Code is available at https://github.com/yuhuan-wu/GAPNet.

Keywords: Salient object detection, lightweight model, granularity-aware paradigm, multi-scale feature fusion

1 Introduction

Salient object detection (SOD) aims to detect the most salient region of interest in images by approximating the human visual system [1, 2]. Accurate SOD can benefit a variety of vision tasks, including visual tracking [3, 4], semantic segmentation [5, 6], image editing [7], medical imaging [8], and robot navigation [9]. Early SOD methods relied on hand-crafted low-level features that captured object details and boundaries but lacked high-level semantics [10], resulting in suboptimal object localization.

Recently, the performance of SOD tasks has been significantly improved by applying Convolutional Neural Networks (CNNs) that can learn low-level features at the bottom layers and high-level features at the top layers [11]. Current state-of-the-art regular models [12–19] made several significant successes in recent years. These models primarily utilize established network architectures [20–24], which can extract very powerful pretrained features. However, these models incur substantial computational overhead, hindering deployment on energy-constrained edge devices.

Notably, these constraints have sparked growing interest in lightweight SOD. However, existing lightweight models, such as EDN-Lite [18] and SAMNet [25], face challenges in achieving comparable performance to heavyweight counterparts due to their use of lightweight backbones like EfficientNet-B0 [26] and MobileNet-V2 [27]. These backbones often compromise multi-level feature representation capabilities, leading to reduced accuracy. To differentiate our work, we redesign the decoder to exploit the limited feature richness of lightweight backbones more effectively. Instead of merely contrasting with heavyweight models, we show how our approach augments lightweight representations to narrow the performance gap.

We illustrate popular SOD decoders in Fig. 1(a) and Fig. 1(b). Early methods [28, 29] (Fig. 1(a)) use late fusion strategies, which directly conduct the prediction from the (fused) features from one or multiple stage(s). These decoders are very efficient due to simple architectures, but come with less effective performance. Recently, U-Net styles (Fig. 1 (b)) are more popular in SOD and have been adopted by many approaches [15, 17]. Through top-down feature fusion with deep supervision, they delve into multi-scale low-level and high-level feature learning, which is essential to achieve high performance. However, U-Net-based decoders are not specifically tailored for lightweight models, leading to inefficiencies in leveraging multi-level features and suboptimal performance when deployed on limited-resource platforms.

Based on the above observations, we propose an encoder-decoder structure, as shown in Fig. 1(c), with granularity-aware paradigm (GAPNet) tailored to lightweight SOD. First, we introduce **G**ranularity-**A**ware **C**onnections to refine the low-level and high-level features separately, which are supervised by the non-center and center ground-truths, respectively. Then, an efficient cross-scale global guidance is incorporated to ensure the accurate localization of salient objects at each fusion stage. To enable effective low-level

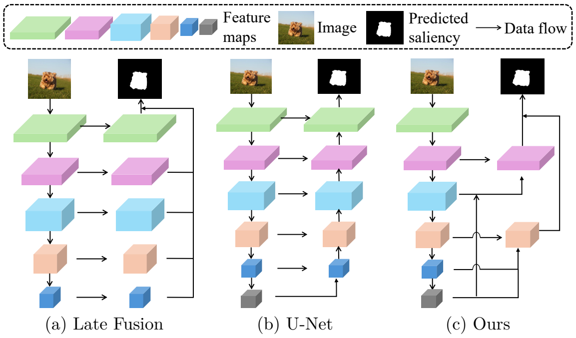


Fig. 1 Different encoder-decoder architectures. (a) Late-fusion decoder side-output is calculated with corresponding encoder features only. Intermediate side-outputs are aggregated to generate the final output. (b) U-Net decoder side-output is calculated with encoder features and higher-level decoder features or global features in a progressive top-down manner. (c) Ours GAPNet fuses global features with low-level and high-level encoder features to compute side-outputs which are then fused as the final output. The high-level and low-level side-outputs have low granularity and high granularity, respectively.

feature fusion at the bottom side, a granular pyramid convolution module with attention refinement (GPC) is constructed to enhance global perception. For high-level feature fusion, we build an efficient cross-scale attention block (CSA) to replace traditional CNN modules. Since the spatial dimensions of high-level features are very low, adopting an attention block in our lightweight model is computationally efficient. Compared to other styles, our proposed granularity-aware connections more effectively optimize the utilization and semantic interpretation of features at each stage, as well as employing targeted supervisions to optimize the performance.

The key novelty and main contributions of this paper are twofold:

- We introduce a granularity-aware paradigm for lightweight image/video SOD that couples scale-specific connections with matching supervision: high-level features learn from coarse object cues, while low-level features are guided by fine boundaries, yielding maximal feature reuse and coherent semantics throughout the pipeline.
- We implement the paradigm with two compact fusion blocks: Granular Pyramid Convolution (GPC) that enriches low-level features via multi-scale aggregation and attention, and Cross-Scale Attention (CSA) that injects

global context into high-level features. Coupled with a lightweight global-attention head, they narrow the accuracy gap to heavyweight models while remaining edge-friendly.

2 Related Work

Salient object detection (SOD) is one of the most significant tasks in computer vision, which can benefit many popular areas like visual tracking [3, 4],, image editing [7], medical imaging [30–32], and camouflaged object detection [33, 34]. In the field of SOD, early popular methods were based on hand-crafted features [35–40]. Deep-learning methods have since dominated SOD owing to their strong generalization across diverse scenarios. The literature categorizes SOD methods into regular, lightweight, and extremely lightweight models. We also review recent advances in encoder-decoder structures and multi-scale fusion. At last, we introduce recent advances of video SOD.

Regular models. Traditional SOD models rely on complex network structures and usually require high computing resources for deployment. The encoder-decoder structure has dominated SOD models where a heavy backbone is used to encode multi-scale features and a decoder is then deployed to fuse these features [16, 18, 41–48]. On top of the encoder, some recent works [18, 49–52] adopt additional CNN modules to extract global features to further improve the performance. In general, heavyweight models achieve high detection accuracy at the cost of low model efficiency.

Lightweight models. Some works build lightweight SOD models with efficient feature fusion modules and lightweight backbones. CSNet has only 100k parameters and is free of pre-training on ImageNet. However, the estimation accuracy is not comparable to large models. Liu *et al.* [53] proposed an efficient HVP module that emulates the primate visual cortex for hierarchical perception learning and builds HVPNet with 1.2M parameters. SAMNet that encodes multi-scale features with a small network is developed in [25]. Fang *et al.* [54] presents lightweight DNTDF with EfficientNet-B0 backbone where PCSP is constructed to enhance the propagation of high-level features during decoding. Wu *et al.* [18] proposed an extremely downsampled module on top of the encoder to extract global features and build an effective decoder to recover object details from the global features. The lightweight version EDN-Lite adopts MobileNet-V2 as the backbone and refreshes state-of-the-art lightweight performance significantly. ADMNet [55] achieved near–heavyweight accuracy by fusing multi-scale context via a compact perception block and sharpening predictions with a dual-attention decoder. Overall, lightweight models sacrifice detection accuracy for lower requirements for computing resources.

Recently, some studies propose extremely lightweight SOD models, exhibit several times fewer parameters than recent lightweight models. For example, CSNet [56] introduced a generalized OctConv block as the basic module for cross-stage multi-scale feature fusion. In [57], the wavelet transform fusion module (WTFM) is built by introducing the wavelet transform theory to CNNs and then used to construct the extremely lightweight model ELWNet which has only 76K parameters. Recently, LARNet and its variant LARNet* are built tailored to lightweight SOD [58]. The newly designed context gating module (CGM) proficiently enhances the features at all levels by transmitting global information. Although the above methods are superior in terms of the model size, their FLOPs and throughput remain comparable to lightweight methods, and their accuracy still lags significantly behind.

Encoder-decoder structures. Many SOD models adopt the encoder-decoder structure to effectively learn multi-level multi-scale features [47, 59–62]. The encoder extracts features from the original image and the decoder integrates these features to the full saliency map using different manners. The architectures include late fusion [63, 64], its variant CPD [65], U-Net [18, 25] and its variants DNA [66] and CTD [50, 52]. The basic late fusion and U-Net architectures are illustrated in Fig. 1(a) and Fig. 1(b), respectively. The former method generates the final output with late fusion and is more efficient due to its simple structure. The latter method fuses features in a top-down manner and is more effective. However, the semantics could be easily affected by low-level features through progressive top-down feature fusion. To take advantage of intermediate decoder features, a deep supervision mechanism [67] has

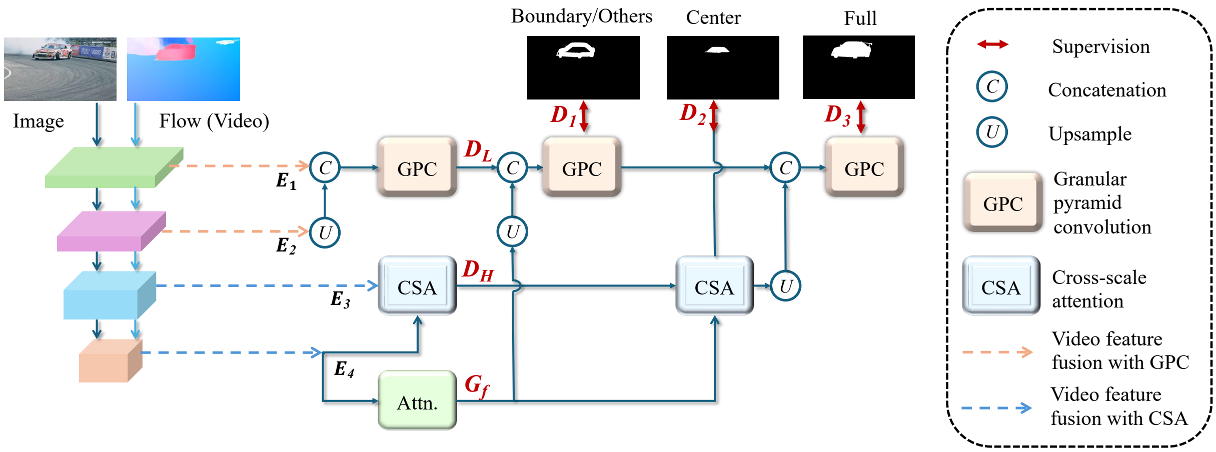


Fig. 2 **Structure of the proposed network**. GT: ground truth. The first layer of backbone is not shown in this figure. GPC is used for fusion of low-scale features and CSA is used for fusion of high-level features. Low-scale side-output D_1 is supervised by boundary/others saliency of high granularity while high-scale side-output D_2 is supervised by center saliency of low granularity. Final output D_3 is supervised by the full saliency map.

been applied to improve the performance of SOD. Existing works [18, 25, 63, 64, 66] utilize the full saliency map to supervise side-outputs of different scales, introducing a significant performance improvement. However, smaller features must be heavily upsampled, making uniform supervision suboptimal.

Although LDF [68] contributed a label decoupling framework by decomposing saliency labels into body and detail maps, this framework relies on iterative feature interactions and multiple training stages to refine predictions for heavy-weight models. Instead of iterative refinement, we introduce a granularity-aware supervision mechanism tailored to lightweight models within a single training stage. By directly assigning boundary-level guidance to low-level features and coarse object-level guidance to high-level features, our approach aligns supervision granularity with each decoder stage without relying on iterative feature interaction.

Moreover, most methods employ U-Net-based decoders [18, 68–71], which are not initially tailored for lightweight models. This leads to inefficiencies in leveraging multi-level features and suboptimal performance on resource-limited platforms. Instead, we propose a simpler and more direct decoder with granularity-aware connections that does not rely on complex iterations. This design establishes a new lightweight-centric paradigm by matching each feature scale with appropriate supervisory signals. Our GPC and CSA modules, carefully devised for multi-scale feature fusion under lightweight constraints, help achieve balanced, effective, and resource-friendly SOD modeling on resource-limited devices.

SOD in the video domain. In contrast to image-based SOD, video SOD generally incorporates the modeling of spatiotemporal features to capture both spatial appearance and temporal consistency across frames [72–79]. For example, TENet [80] employed the GT, the learnable prediction, and their weighted sum as an attention map. The weights gradually shift toward emphasizing the prediction as training progresses, thereby increasing the segmentation difficulty and improving spatial feature learning. FSNet [75] introduced a cross-attention which is computed between motion features and appearance features, enabling effective feature fusion that is subsequently used for salient object prediction. DCFNet [81] proposed to leverage the two adjacent frames of the current frame as temporal attention to guide information propagation. By employing matrix multiplication, it diffuses contextual cues throughout the entire spatial domain, achieving a dynamic filtering strategy with an effectively enlarged receptive field. MMN [82] applied two neighboring frames of the current frame as the memory to guide the extraction of high-level semantic features. This facilitates the integration of temporal information across frames, thereby enhancing the model's ability to accurately identify salient object characteristics. Liu *et al.* [83] proposed using optical flow to guide the sampling window positions within input video clips, enabling more effective modeling of the spatial-temporal features of the same object. Li *et al.* [84] grouped keyframes based on background similarity and employed different models to learn each group. Each model focused on a specific type of background, thereby reducing the difficulty of modeling videos with frequent viewpoint changes. Despite the above success, these works are with significant computational cost. Instead, we introduce a lightweight solution that is several times faster than existing heavyweight models and narrows the gap between lightweight and heavyweight models in video SOD.

3 Methodology

In this section, we first provide the details of our network structure in Sec. 3.1. Then, we present our granularity-aware connections for multi-scale feature fusion in Sec. 3.2. Last, we introduce the granularity-aware deep supervision in Sec. 3.3.

3.1 Network Structure

Fig. 2 presents the overall pipeline, which comprises an encoder (Sec. 3.1.1), a global-feature extractor (Sec. 3.1.2), and a decoder (Sec. 3.1.1).

3.1.1 Backbone encoder

Due to computational constraints, we employ the well-known MobileNet-V2 [27] as the backbone. Following previous studies [18, 85], we remove the final pooling and fully connected layers to obtain a fully convolutional network suited to dense prediction. The MobileNet-V2 encoder consists of five stages, with strides of 2, 4, 8, 16, and 32, respectively. The last four stages, denoted as E_1, E_2, E_3, and E_4, are utilized for decoding in our work. These encoder features correspond to scales of $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$, and $\frac{1}{32}$, respectively. For simplicity, the first stage of the encoder is not depicted in the structure. Our framework naturally extends to video sequences by incorporating temporal information through a two-stream architecture. For video inputs, we process both RGB frames and optical flow through separate lightweight backbones, fusing them at multiple hierarchical levels within our granularity-aware connections thereafter.

3.1.2 Global feature extractor

As mentioned previously, the scale of the final encoder outputs is only $\frac{1}{32}$ of the original input image. Incorporating a global feature extractor with vision transformers is efficient at such a small scale. Therefore, we stack a transformer module atop the encoder to extract global features, which are subsequently combined with local features for multi-scale feature fusion. In the following sections, we will detail the global feature extractor.

Firstly, the attention is calculated as:

$$Att_G = E_4 + \text{Attention}(\text{LayerNorm}(E_4)) \tag{1}$$

where $\text{LayerNorm}(\cdot)$ denotes layer normalization. $\text{Attention}(\cdot)$ is the self-attention defined as below:

$$\begin{aligned} (Q, K, V) &= X(W^Q, W^K, W^V) \\ \text{Attention}(X) &= \text{Linear}(\text{softmax}(\frac{QK^T}{\sqrt{d_k}})V) \end{aligned} \tag{2}$$

where the input features are flattened with the spatial dimension, $\text{Linear}(\cdot)$ denotes one linear transformation layer, d_k is the scaling factor of the attention.

Then, an inverted residual block (IRB) [27] is applied as the feed-forward network (FFN) to compute the global features G_f, formulated as

$$G_f = Att_G + \text{IRB}(\text{LayerNorm}(Att_G)) \tag{3}$$

3.1.3 Decoder network

In our GAPNet, the hierarchical decoder incorporates five feature fusion modules. To maintain the efficiency of our framework, we have developed two types of modules for feature fusion in the decoder: granular pyramid pooling convolution (GPC) and cross-scale attention (CSA). These modules are designed to fuse low-level and high-level features, respectively. We will provide further details on these modules in Sec. 3.2. As depicted in Fig. 2, low-level encoder features E_1 and E_2 are decoded to D_L, and high-level encoder features E_3 and E_4 are decoded to D_H, as calculated below:

$$\begin{aligned} D_L &= \mathcal{H}_P(\text{Concat}(\text{Upsample}(\mathcal{G}(E_2)), \mathcal{G}(E_1))) \\ D_H &= \mathcal{H}_C(\text{Concat}(\mathcal{G}(E_3), \mathcal{G}(E_4))), \end{aligned} \tag{4}$$

where $\mathcal{H}_P(\cdot)$ and $\mathcal{H}_C(\cdot)$ are the GPC and CSA modules, respectively. $\mathcal{G}(\cdot)$ denotes a convolution followed by batch normalization and ReLU activation. Upsample$(\cdot)$ upsamples low-scale features to the same resolution as high-scale features using bilinear interpolation. For the concatenation of $\mathcal{H}_C(\cdot)$, it is not necessary to upsample the low-scale features because the spatial features are flattened into a vector before the concatenation.

Then, the decoder features D_L and D_H are fused with the global features G_f to calculate low-level side-output D_1 and high-level side-output D_2, expressed as

$$\begin{aligned} D_1 &= \mathcal{H}_{\mathcal{P}}(\text{Concat}(\text{Upsample}(G_f), D_L)) \\ D_2 &= \mathcal{H}_{\mathcal{C}}(\text{Concat}(D_H, G_f)), \end{aligned} \tag{5}$$

Last, the final decoder output is computed by fusing the side-outputs D_1 and D_2, shown as

$$D_3 = \mathcal{H}_{\mathcal{P}}(\text{Concat}(\text{Upsample}(D_2), D_1)), \tag{6}$$

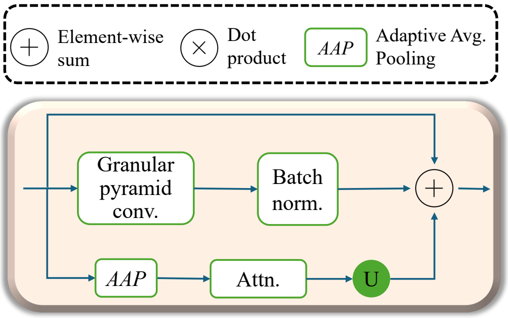
we employ GPC for the final fusion because it excels at preserving fine-grained boundary details at high spatial resolutions, which is essential for accurate final predictions. Additionally, GPC is computationally more efficient than CSA when processing the high-resolution concatenated features.

3.2 Multi-scale Feature Fusion

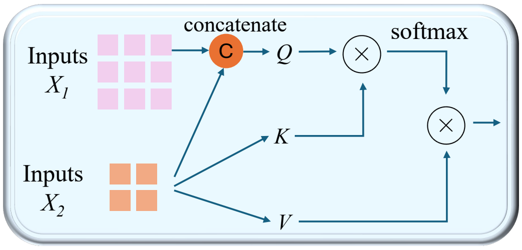
Successful salient object detection necessitates simultaneous global localization [18] and multi-scale feature learning [16]. Effectively extracting both, while maintaining efficiency, presents a significant challenge due to computational constraints. In response, we have developed two distinct strategies: CNN-based (Sec. 3.2.1) and transformer-based (Sec. 3.2.2) modules, designed specifically for low-level and high-level feature fusion, respectively. Further details of these modules are discussed below.

3.2.1 Granular pyramid convolution with efficient self-attention

We introduce an efficient GPC module for low-scale feature fusion as shown in Fig. 3(a). It consists of a multi-scale feature extraction branch and an efficient global attention. For the global



(a) Granular pyramid convolution with efficient attention.



(b) Cross-scale attention.

Fig. 3 Illustration of GPC and CSA for multi-scale feature fusion. For cross-scale attention, Q is computed with combined X_1 and X_2 while K, V are computed with X_2 only.

attention module, we first apply adaptive average pooling to downsample the input to $m \times m$, thereby reducing computational overhead. Attention is computed on the downsampled feature and then upsampled via bilinear interpolation. The whole process can be elaborated as below:

$$\begin{aligned} F_{ds} &= AAP(F_{in}, (m \times m)) \\ Att_P &= F_{ds} + \text{Attention}(\text{LayerNorm}(F_{ds})) \\ F_{out}^{\text{A}} &= \text{Upsample}(Att_P) \end{aligned} \tag{7}$$

where AAP is a 2D adaptive average pooling layer that pools the input feature to the size of $m \times m$. Attention$(\cdot)$ is the vanilla attention shown in Eq. (2). Upsample$(\cdot)$ upsamples attention to the same size as the input features F_{in}.

For the CNN block, the input features F_{in} are first split into four feature maps along the channel dimension, denoted as F_1, F_2, F_3 and F_4. Unlike recent approaches [18, 86] that evenly split channels we allocate ratios of 1/8, 1/8, 1/4, and 1/2 so that smaller dilation rates are applied to high-scale features. We concatenate the features of each split followed by a 1×1 convolution. The above processes are elaborated as below:

$$\begin{aligned} C_i &= \mathrm{Conv}_{3\times3}^{a_i}(F_i), \quad i \in \{1,2,3,4\} \\ F_{out}^{\mathrm{C}} &= \mathrm{Conv}_{1\times1}(\mathrm{Concat}(\mathrm{C}_1,\mathrm{C}_2,\mathrm{C}_3,\mathrm{C}_4)) \end{aligned} \tag{8}$$

where $\mathrm{Conv}_{3\times3}^{a_i}(\cdot)$ is a 3×3 atrous convolution with an atrous rate of a_i followed by batch normalization.

Finally, we add a residual connection to aggregate the output feature F_{out}, which is computed as

$$F_{out} = F_{out}^{\mathrm{A}} + F_{out}^{\mathrm{C}} + F_{in} \tag{9}$$

3.2.2 Cross-scale attention mechanism

For high-level features, the spatial resolution is significantly reduced compared to the original image, which enables the deployment of attention mechanisms even with limited computational resources. Consequently, we have developed a CSA block for high-level feature fusion, as illustrated in Fig. 3(b). Unlike traditional attention mechanisms that first concatenate input features of different scales and then compute Q, K, and V, our cross-level attention approach computes Q using combined input features, while K and V are derived solely from high-level features. This approach is formulated as follows:

$$\begin{aligned} Q &= \mathrm{Concat}(X_1, X_2)W^Q \\ (K, V) &= X_2(W^K, W^V) \end{aligned} \tag{10}$$

where X_1, X_2 are the flattened low-level and high-level features, respectively. LayerNorm($\cdot$) is performed before calculating Q, K and V.

This cross-scale attention mechanism significantly reduces the computational burden. In Eq. (4), the scales of E_3 and E_4 are $\frac{1}{16}$ and $\frac{1}{32}$, respectively. Consequently, the scale X_2 constitutes only one fifth of $\mathrm{Concat}(X_1, X_2)$, reducing the complexity of the cross-scale attention to just $\frac{1}{25}$ of that observed in vanilla attention mechanisms. Similarly to standard transformer blocks, attention is computed as outlined in Eq. (2). Finally, an FFN comprising two linear layers with a residual connection is deployed to compute the fused features.

3.2.3 Video feature fusion

For video salient object detection, our framework adopts a two-stream architecture that processes both RGB frames and optical flow information to capture spatial-temporal dependencies. The fusion of RGB and optical flow features occurs at multiple hierarchical levels within our granularity-aware connections.

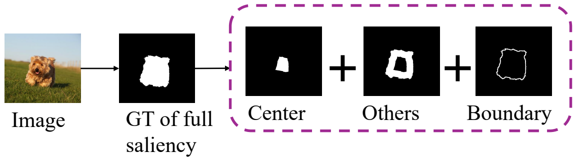


Fig. 4 Illustration of decomposing the foreground of full saliency map into multi-granularity regions: center, boundary and others. Black and white regions represent background and foreground, respectively.

At low-level stages (E_1 and E_2), we employ a simple yet effective fusion strategy that combines additive and multiplicative attention mechanisms before applying the granular pyramid convolution. Specifically, the optical flow features are first passed through a sigmoid activation to generate attention weights, which are then used to modulate the RGB features through element-wise multiplication. The final fused features combine both the attention-modulated RGB features and the original features from both modalities. This fusion mechanism allows the optical flow to serve as an attention gate that highlights motion-relevant regions while preserving complementary information from both streams.

At high-level stages (E_3 and E_4), we leverage the same cross-scale attention (CSA) modules used in our granularity-aware connections. The CSA mechanism naturally accommodates the fusion of multi-modal features by treating RGB and optical flow features as different input sequences. The CSA module computes cross-attention between RGB and flow features, enabling the model to capture long-range temporal dependencies and motion-guided spatial attention.

This hierarchical fusion strategy aligns with our granularity-aware paradigm: low-level fusion preserves fine-grained motion details essential for accurate boundary delineation, while high-level fusion captures coarse temporal semantics for robust object localization. The fused features are then processed through the same decoder structure as described in Sec. 3.1.3, maintaining computational efficiency while enhancing temporal consistency in video salient object detection.

3.3 Granularity-aware Deep Supervision

Based on various encoder-decoder structures depicted in Fig. 1, a deep supervision mechanism can be employed to leverage decoder side-outputs effectively. Existing methods such as HED [63], U-Net [18], and variants of U-Net like CPD [65], typically utilize the full saliency map to supervise side-outputs at different scales. Some recent methods employ the edge supervision [14] in the low-level features or apply label decoupling strategy with an iterative training strategy. In contrast, our approach, as illustrated in Fig. 2, proposes using distinct ground truths (center, edge, others, full) to supervise different outputs in a single stage, enhancing the specificity and effectiveness of the training process for lightweight SOD.

3.3.1 Decomposition of ground-truth saliency map

According to the Euclidean distance to the nearest background pixel, each pixel of the saliency foreground is classified into three regions [18]: the boundary, which is close to the background; the center, which is far from the background; and others, which are located in the middle of an object. Specifically, the boundary region comprises pixels that are less than five pixels away from the closest background pixel. Pixels that rank in the top 20% in terms of distance from the nearest background pixel constitute the center region. Any foreground pixels that do not qualify for inclusion in either the boundary or center regions are categorized into the other region. The aggregation of the center and other regions is referred to as the boundary-others region. The center region represents the abstract location of the object, while the boundary delineates the fine-grained edges of the object. For illustration, an example is provided in Fig. 4.

Based on this classification, we employ low-granularity center saliency to supervise the high-level side-output, and high-granularity boundary and others saliency to supervise the low-level side-output. The final output is supervised using the full saliency map.

3.3.2 Loss function

The loss function combines the binary cross-entropy loss and Dice Loss [87], defined as

$$\begin{aligned}\mathcal{L}_{bce} &= -G\log P - (1-G)\log(1-P)\\ \mathcal{L}_{dice} &= 1 - \frac{2\cdot G\cdot P}{G+P}\\ \mathcal{L} &= \mathcal{L}_{bce} + \mathcal{L}_{dice}\end{aligned} \tag{11}$$

where P and G denote the predicted and ground-truth saliency map, respectively. "$\cdot$" operation is the dot product. $\cdot$ denotes the ℓ_1 norm. L_{bce}, L_{dice} and L represent the binary cross-entropy loss, dice loss and combined loss, respectively. The Dice loss is an effective way to address class-imbalance datasets.

There are two side-outputs and one final output and the overall loss that we use for training is computed as

$$\mathcal{L}_{overall} = \sum_{i=1}^{3}\mathcal{L}(P_i, G_i) \tag{12}$$

where G_1, G_2 and G_3 are the ground-truth boundary-others saliency map, center saliency map and full saliency map, respectively. P_i is the corresponding predicted saliency map calculated from decoder side-outputs D_1, D_2 and D_3 in Eq. (5) and Eq. (6), shown as

$$P_i = \sigma(\mathrm{Upsample}(\mathrm{Conv}_{1\times1}(D_i))), \quad i\in\{1,2,3\} \tag{13}$$

where $\mathrm{Conv}_{1\times1}(\cdot)$ denotes a convolutional layer without normalization and activation. $\mathrm{Upsample}(\cdot)$ upsamples input features to the same resolution as the full saliency map using bilinear interpolation. $\sigma(\cdot)$ is the standard sigmoid function.

4 Experiments

4.1 Experimental Setup

Implementation details. The proposed model is implemented in PyTorch [89] with a single NVIDIA RTX3090 GPU. Training is carried out over 30 epochs using the Adam optimizer [90], with parameters set to $\beta_1 = 0.9$, $\beta_2 = 0.99$, a weight decay of 10^{-4}, and a batch size of 32. We

Table 1 Comparison of GAPNet with state-of-the-art heavyweight and lightweight SOD methods. The best performance in each row among lightweight models is highlighted in bold.

Method		Heavyweight models (# Param > 20 M)										Lightweight models (# Param < 2 M)							
		CPD	PoolNet	ITSD	MINet	VST	CTD	ICON	EDN	SRF	PiNet	HVPNet	CSNet	SAMNet	EDN-Lite	ELWNet	LARNet	ADMNet+	Ours
# Reference		[65]	[15]	[17]	[16]	[88]	[50]	[19]	[18]	[51]	[46]	[53]	[56]	[25]	[18]	[57]	[58]	[55]	-
# Param (M)		47.85	68.26	26.47	162.38	44.56	24.64	33.09	42.85	91.59	27.20	1.24	0.14	1.33	1.80	0.07	0.09	0.84	1.99
FLOPs (G)		17.74	89.06	15.95	87.02	41.36	12.35	20.91	20.37	21.77	–	1.13	0.57	0.54	1.02	0.38	0.82	3.30	1.26
Speed (FPS)		60	83	77	62	73	148	89	77	39	–	749	675	459	652	–	–	115	571
DUTS-TE	$F_\beta^{\max}$	0.865	0.874	0.882	0.880	0.890	0.896	0.891	0.893	0.913	0.865	0.837	0.804	0.833	0.856	–	–	0.840	**0.867**
	F_β^w	0.794	0.806	0.822	0.824	0.827	0.846	0.835	0.844	0.871	0.817	0.730	0.643	0.729	0.789	–	–	0.767	**0.804**
	MAE	0.043	0.040	0.041	0.038	0.038	0.034	0.037	0.035	0.027	0.041	0.058	0.075	0.058	0.045	0.075	0.069	0.052	**0.042**
	S_α	0.869	0.883	0.884	0.883	0.896	0.892	0.888	0.892	0.910	0.864	0.849	0.822	0.849	0.862	–	0.820	0.849	**0.872**
	$E_\xi^{\max}$	0.914	0.923	0.930	0.927	0.939	0.935	0.932	0.934	0.952	0.911	0.899	0.875	0.902	0.910	–	–	0.892	**0.922**
	E_ξ^{mean}	0.898	0.904	0.914	0.917	0.919	0.929	0.923	0.925	0.943	0.906	0.860	0.820	0.860	0.895	–	–	0.882	**0.910**
DUT-OMRON	$F_\beta^{\max}$	0.797	0.792	0.818	0.795	0.822	0.818	0.821	0.821	0.825	0.793	0.796	0.761	0.795	0.783	–	–	0.797	**0.806**
	F_β^w	0.719	0.729	0.750	0.738	0.755	0.762	0.761	0.770	0.784	–	0.700	0.620	0.699	0.721	–	–	0.729	**0.738**
	MAE	0.056	0.055	0.061	0.056	0.058	0.052	0.057	0.050	0.043	0.055	0.064	0.080	0.065	0.058	0.083	0.080	0.058	**0.057**
	S_α	0.825	0.836	0.840	0.833	0.850	0.844	0.844	0.849	0.861	0.821	0.831	0.805	0.830	0.824	–	0.797	0.826	**0.833**
	$E_\xi^{\max}$	0.868	0.871	0.880	0.869	0.888	0.881	0.884	0.885	0.894	0.863	0.876	0.853	0.877	0.860	–	–	0.869	**0.876**
	E_ξ^{mean}	0.847	0.854	0.865	0.860	0.871	0.875	0.876	0.878	0.884	0.859	0.839	0.801	0.841	0.848	–	–	0.857	**0.866**
HKU-IS	$F_\beta^{\max}$	0.925	0.930	0.934	0.934	0.942	0.940	0.939	0.940	0.947	0.928	0.914	0.896	0.914	0.922	–	–	0.918	**0.929**
	F_β^w	0.875	0.881	0.894	0.897	0.897	0.909	0.902	0.908	0.915	0.896	0.840	0.777	0.837	0.877	–	–	0.872	**0.889**
	MAE	0.034	0.033	0.031	0.029	0.030	0.027	0.029	0.027	0.024	0.030	0.045	0.060	0.045	0.035	0.051	0.046	0.036	**0.032**
	S_α	0.905	0.915	0.917	0.919	0.928	0.921	0.920	0.924	0.931	0.904	0.899	0.881	0.898	0.906	–	0.883	0.901	**0.914**
	$E_\xi^{\max}$	0.950	0.954	0.960	0.960	0.968	0.961	0.960	0.962	0.969	0.951	0.946	0.933	0.946	0.948	–	–	0.946	**0.957**
	E_ξ^{mean}	0.938	0.939	0.947	0.952	0.952	0.956	0.953	0.955	0.960	0.946	0.914	0.883	0.912	0.936	–	–	0.934	**0.947**
ECSSD	$F_\beta^{\max}$	0.939	0.943	0.947	0.946	0.951	0.949	0.950	0.950	0.957	0.935	0.927	0.912	0.926	0.934	–	–	0.922	**0.938**
	F_β^w	0.898	0.896	0.910	0.911	0.910	0.915	0.918	0.918	0.926	0.902	0.854	0.806	0.858	0.890	–	–	0.871	**0.898**
	MAE	0.037	0.039	0.035	0.034	0.034	0.032	0.032	0.033	0.027	0.039	0.053	0.066	0.051	0.043	0.061	0.055	0.051	**0.040**
	S_α	0.918	0.921	0.925	0.925	0.932	0.925	0.929	0.927	0.936	0.910	0.903	0.893	0.907	0.911	–	0.888	0.900	**0.916**
	$E_\xi^{\max}$	0.951	0.952	0.959	0.957	0.964	0.956	0.960	0.958	0.965	0.948	0.940	0.931	0.944	0.944	–	–	0.933	**0.950**
	E_ξ^{mean}	0.942	0.940	0.947	0.950	0.951	0.950	0.954	0.951	0.957	0.944	0.911	0.886	0.916	0.933	–	–	0.914	**0.941**
PASCAL-S	$F_\beta^{\max}$	0.859	0.862	0.870	0.865	0.875	0.877	0.876	0.879	0.892	0.858	0.838	0.826	0.836	0.852	–	–	0.827	**0.860**
	F_β^w	0.794	0.793	0.812	0.809	0.816	0.822	0.818	0.827	0.848	0.807	0.746	0.691	0.738	0.788	–	–	0.752	**0.793**
	MAE	0.071	0.075	0.066	0.064	0.062	0.061	0.064	0.062	0.051	0.069	0.090	0.104	0.092	0.073	0.102	0.096	0.088	**0.073**
	S_α	0.848	0.849	0.859	0.856	0.872	0.863	0.861	0.865	0.881	0.837	0.830	0.814	0.826	0.842	–	0.810	0.815	**0.843**
	$E_\xi^{\max}$	0.891	0.891	0.908	0.903	0.918	0.906	0.908	0.908	0.928	0.889	0.872	0.860	0.870	0.890	–	–	0.862	**0.890**
	E_ξ^{mean}	0.882	0.880	0.895	0.896	0.902	0.901	0.899	0.902	0.919	0.886	0.844	0.815	0.839	0.878	–	–	0.851	**0.881**

employ a polynomial learning rate scheduler with an initial learning rate of 1.7×10^{-4} and a power of 0.9. The adaptive pooling size of the GPC module is set to $m = 7$ Eq. (7). During training, the input images are resized to 320×320, 352×352, and 384×384 for augmentation purposes. During inference, images are resized to 384×384. The CSA and GPC modules are highly efficient, with just 0.065M and 0.020M parameters. For video SOD, we first train our model using static images of DUTS training set and then finetune on the video dataset. Following previous popular works [75, 83, 91], we apply FlowNet 2.0 [92] to generate the offline optical flows. The video SOD training hyper-parameters match those of image SOD, except that the learning rate is reduced by a factor of ten.

SOD Datasets. The proposed method has been tested on five commonly-used datasets, including three large datasets: DUTS [93], DUT-OMRON [36], HKU-IS [94], and two smaller datasets: ECSSD [95] and PASCAL-S [96]. These datasets comprise 15572, 5168, 4447, 1000, and 850 natural images with corresponding pixel-level labels, respectively. Following methodologies from prior studies [97–99], we train our model on the DUTS training set, which contains 10553 images, and evaluate it on the DUTS test set (DUTS-TE, 5019 images) and the other four datasets.

Video SOD Datasets. We utilize four commonly-used datasets DAVSOD [72], DAVIS [100], SegTrack-V2 [101], and ViSal [102] to construct the experiments. Following other approaches, our model is also trained on the training set of DAVSOD [72] and DAVIS [100], which have 91 clips in total. Other data are for testing. For DAVSOD, we use the easy set of 35 clips for testing.

Evaluation Criteria. We employ six widely-used metrics to evaluate all methods, which include the maximum F-measure score ($F_\beta^{\max}$), weighted F-measure score (F_β^w) [103], mean absolute error (MAE), S-measure (S_α) [104], maximum E-measure ($E_\xi^{\max}$), and mean E-measure (E_ξ^{mean}) [105]. Except for MAE, a higher value indicates better performance for all metrics. F-measure is the weighted harmonic mean of precision and recall and can be calculated as

$$F_\beta = \frac{(1+\beta^2) \times \text{Precision} \times \text{Recall}}{\beta^2 \times \text{Precision} + \text{Recall}} \tag{14}$$

where $\beta^2 = 0.3$ to emphasize the importance of precision, following previous studies [13, 15, 29, 85]. $F_\beta^{\max}$ is the maximum F_β under different binary thresholds. F_β^w solves the problems of F-measure that may cause three types of flaw, *i.e.*, interpolation, dependency, and equal-importance [103].

MAE measures the similarity between the predicted saliency map P and the ground-truth saliency map G, which can be computed as

$$\text{MAE}(P, G) = \frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} \| P_{i,j} - G_{i,j} \| \tag{15}$$

where H and W denote the height and width of the saliency map, respectively.

S-measure (S_α) [104] and E-measure (E_ξ) [105] have been increasingly popular for SOD evaluation recently [16, 58, 106]. S-measure calculates the structural similarity between the predicted saliency map and the ground-truth map. E-measure computes the similarity for the predicted map binarized by different thresholds and the binary ground-truth map. Thus, they are significant alternatives that could provide more comprehensive SOD evaluations. In this paper, we compute the maximum and average E-measures ($E_\xi^{\max}$, E_ξ^{mean}) among all binary thresholds. We use the official codes from [104, 105] to compute the above metrics.

4.2 Experimental Comparisons

Image SOD. We compare our model against nine heavyweight models with over 10M parameters and six lightweight models with no more than 10M parameters. For competing models that offer both ResNet-50 and VGG-16 backbones, the ResNet-50 backbone is utilized. For lightweight models, all models are with the MobileNetV2 backbone, except that CSNet, ELWNet, and LARNet designed their backbones for extremely lightweight SOD. For a fair comparison, we use the saliency maps provided by the official repositories of the benchmarking methods and use the same code for evaluation. To assess model efficiency, we re-implement these models on the same workstation equipped with a single NVIDIA RTX3090 GPU. The input image sizes for the competing models adhere to the default settings specified in their original publications.

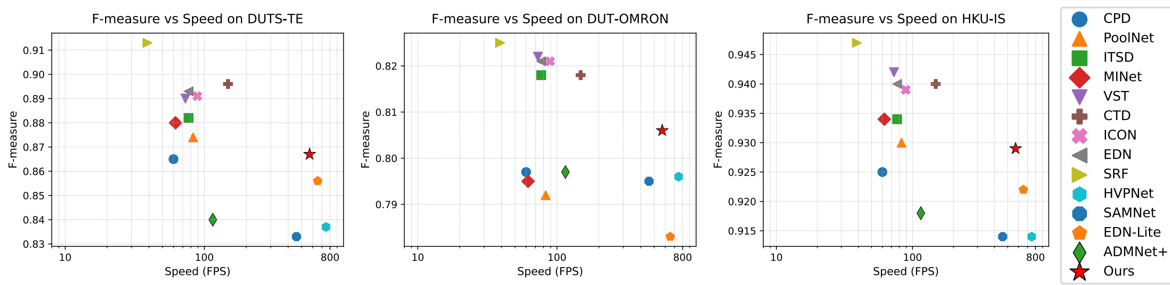
For LARNet [58] and ELWNet [57], where no official codes or saliency maps are available, we directly extract data on the number of parameters, FLOPs, and selected performance metrics from the published papers.

Video SOD. We compare our model against several recent heavyweight models. For fair comparison, we re-implement two recent strongest heavyweight models [82, 91] in the lightweight setting, *i.e.*, replacing the backbone and 3×3 convolutions with MobileNetV2 and 3×3 depthwise convolutions, respectively. Following previous popular works [72], we apply S-measure, maximum F-measure, and MAE as the evaluation metrics.

4.2.1 Quantitative comparison

Image SOD. A comprehensive quantitative comparison of our model with competing methods is presented in Table 1. Our model consistently outperforms or matches other lightweight methods across the five datasets using all six metrics. Specifically, our model surpasses the state-of-the-art lightweight model EDN-Lite [18] by margins of 1.1%, 2.3%, 0.7%, 0.4%, and 0.8% in terms of $F_\beta^{\max}$ across the datasets. Moreover, using E_ξ^{mean}, our model achieves performance improvements of 1.5%, 1.8%, 1.1%, 0.8%, and 0.3%. For S_α, our model beats state-of-the-art by 1.0%, 0.9%, and 0.8% on the three large datasets, namely DUTS-TE, DUT-OMRON, and HKU-IS. Notably, the most significant improvements are observed on the DUT-OMRON dataset, where S_α, $E_\xi^{\max}$, E_ξ^{mean}, $F_\beta^{\max}$, F_β^w, and MAE are improved by 0.9%, 1.6%, 1.8%, 2.3%, 1.7%, and 0.1%, respectively.

In terms of model efficiency, our model possesses more parameters and is comparatively



(a) DUTS-TE [93] (b) DUT-OMRON [36] (c) HKU-IS [94]

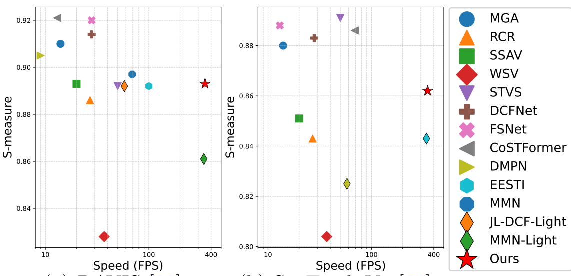
Fig. 5 Speed and accuracy comparison with state-of-the-art SOD methods. Our model outperforms all the lightweight models and some of the heavyweight models. Inference speed is plotted using logarithm with base 10.

Table 2 Comparison with state-of-the-art methods on video SOD. The best performance of lightweight models is marked in bold.

Method	Param (M)	FPS	DAVIS [100]			ViSal [102]			DAVSOD [72]			SegTrack-V2 [101]		
			S_α	F_β^{max}	MAE	S_α	F_β^{max}	MAE	S_α	F_β^{max}	MAE	S_α	F_β^{max}	MAE
Heavyweight models														
MGA [107]	87.5	14	0.910	0.892	0.022	0.940	0.936	0.017	0.741	0.643	0.083	0.880	0.829	0.027
RCR [108]	51.5	27	0.886	0.848	0.027	0.922	0.906	0.027	0.741	0.653	0.087	0.843	0.782	0.035
SSAV [72]	59.0	20	0.893	0.861	0.028	0.943	0.939	0.020	0.724	0.603	0.092	0.851	0.801	0.023
LTSD [73]	–	–	0.897	0.891	0.021	–	–	–	0.768	0.689	0.075	0.880	0.866	0.018
TENet [80]	–	–	0.905	0.894	0.021	0.943	0.947	0.021	0.753	0.648	0.078	–	–	–
WSV [109]	33.0	37	0.828	0.779	0.037	0.857	0.831	0.041	0.705	0.605	0.103	0.804	0.738	0.033
STVS [74]	46.0	50	0.892	0.865	0.023	0.954	0.953	0.013	0.744	0.650	0.086	0.891	0.860	0.017
DCFNet [81]	68.5	28	0.914	0.900	0.016	0.952	0.953	0.010	0.741	0.660	0.074	0.883	0.839	0.015
FSNet [75]	97.9	28	0.920	0.907	0.020	–	–	–	0.773	0.685	0.072	–	–	–
CoSTFormer [83]	–	13	0.921	0.903	0.014	–	–	–	0.806	0.731	0.061	0.888	0.833	0.015
DMPN [79]	152.2	9	0.905	0.888	0.021	0.929	–	0.016	0.755	0.655	0.069	–	–	–
EESTI [74]	46.2	100	0.892	0.865	0.023	0.952	0.952	0.013	0.746	0.651	0.086	0.891	0.860	0.017
Li *et al.* [84]	–	–	0.906	0.888	0.018	–	–	–	0.777	0.716	0.072	–	–	–
MMN [82]	49.0	69	0.897	0.877	0.020	0.947	0.948	0.012	0.777	0.708	0.065	0.886	0.850	0.014
Lightweight models														
JL-DCF-Light [91]	2.1	58	0.892	0.863	0.025	0.882	0.858	0.038	**0.728**	**0.630**	**0.088**	0.825	0.743	0.030
MMN-Light [82]	3.1	340	0.861	0.822	0.025	0.884	0.864	0.035	0.700	0.593	0.089	0.843	0.786	0.023
Ours	3.8	349	**0.893**	**0.864**	**0.021**	**0.886**	**0.867**	**0.033**	0.706	0.597	0.089	**0.862**	**0.804**	**0.021**

less efficient against extremely lightweight models CSNet [56], LARNet [58], and ELWNet [57]. However, there is a notable accuracy gap between these models and ours. For instance, the MAE of LARNet [58] on DUTS-TE is 0.069, whereas our model achieves an MAE of 0.042. Compared to lightweight models with similar parameters, including SAMNet [25], HVPNet [53], and EDN-Lite [18], our model exhibits slightly higher FLOPs and reduced inference speed. This increased computational demand is attributed to the attention modules integrated into our model. In summary, our model establishes new benchmarks in state-of-the-art performance for lightweight SOD models across all test cases, albeit at the expense of marginally higher computational overhead and slower inference speeds.

A comparison of the accuracy ($F_\beta^{\max}$) and inference speed across three large datasets (e.g., DUTS-TE, DUT-OMRON, and HKU-IS) is depicted in Fig. 5. It is evident that our model consistently surpasses other lightweight models across all datasets with significant improvements. In certain test cases, our model achieves performance comparable to or even surpassing some heavyweight models, which exhibit considerably



(a) DAVIS [93] (b) SegTrack-V2 [36]

Fig. 6 Speed and accuracy comparison with state-of-the-art video SOD methods.

slower inference speeds. For example, on the DUT-OMRON dataset, our model outperforms heavyweight models such as CPD [65], PoolNet [15], and MINet [16].

Video SOD. Results are shown in Table 2. We list a comparison of popular heavyweight and lightweight models in recent years. From the results, we can find that our model outperforms the recent lightweight versions of JL-DCF [91] and MMN [82]. Although the lightweight JL-DCF achieves better performance than our model on the DAVSOD dataset, it falls short on other datasets, particularly on SegTrack-V2. Furthermore, our GAPNet operates significantly faster than the lightweight JL-DCF, highlighting its suitability for lightweight applications, especially on edge devices. Compared to heavyweight models, our GAPNet demonstrates competitive performance while offering substantial efficiency advantages. Despite having significantly fewer parameters than heavyweight counterparts, our method operates at much higher inference speeds and achieves comparable or superior accuracy on most datasets especially DAVIS and SegTrack-V2. This demonstrates that our granularity-aware paradigm effectively bridges the performance gap between lightweight and heavyweight approaches, making it highly suitable for real-time applications and resource-constrained environments without sacrificing detection quality.

Following the SOD part, we also illustrate the speed-accuracy comparison as shown in Fig. 6. Our method consistently occupies the upper-right corner of the accuracy-speed plots on both DAVIS and SegTrack-V2, delivering S-measure scores that rival or surpass heavyweight competitors while running an order of magnitude faster (300 FPS). This clear dominance in the speed–accuracy Pareto front highlights the superior efficiency of the proposed framework over all lightweight baselines and many heavyweight models alike.

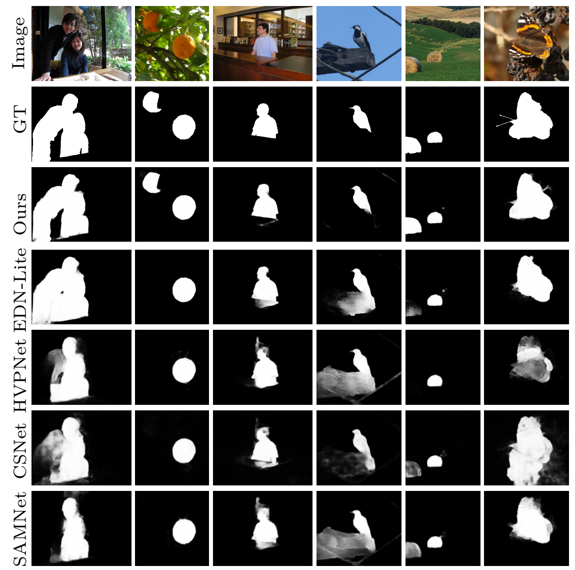


Fig. 7 Qualitative comparison with other lightweight models on image SOD.

4.2.2 Qualitative comparison

A qualitative comparison is illustrated in Fig. 7. It is apparent that our model can accurately identify salient objects with clear boundaries and high confidence, even in complex scenarios. Particularly in images—such as the last two in the figure—where the foreground salient object blends with the background, competing models often incorrectly classify nearby background elements as part of the foreground. In contrast, our model maintains precise segmentation, demonstrating its robustness and accuracy.

4.3 Ablation Study

To demonstrate the efficacy of various modules within our model, as well as the impact of different deep supervision combinations, we conducted an ablation study using the DUTS-TE dataset. This study utilized efficiency metrics and selected performance metrics: $F_\beta^{\max}$, F_β^w, and MAE.

4.3.1 Attention module in GPC

The experimental results concerning the attention module of GPC are summarized in Table 3. It

Table 3 Ablation study of the output dimension of the pooling layer in GPC.

Method	#Param (M)	FLOPs (G)	Speed (FPS)	$F_\beta^{\max}$	F_β^w	MAE
w/o Attn.	1.96	1.25	609	0.864	0.801	0.043
$m = 1$	1.99	1.26	578	0.864	0.800	0.043
$m = 3$	1.99	1.26	572	0.865	0.803	0.043
$m = 7$	1.99	1.26	571	**0.867**	**0.804**	**0.042**
$m = 28$	1.99	1.47	408	0.865	0.803	**0.042**

Table 4 Effect of the split ratios of pyramid convolution module.

Method	Split Ratios	$F_\beta^{\max}$	F_β^w	MAE
Identical Split [18]	$[\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}]$	0.863	0.801	0.043
Ours	$[\frac{1}{8}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}]$	**0.867**	**0.804**	**0.042**

is important to note that m represents the output dimension of the adaptive pooling layer in Eq. (7), and the attention module was evaluated with four different pooling sizes, as well as without the module for comparison. The findings indicate that the attention module enhances estimation accuracy with a minimal increase in the number of parameters, particularly when the pooling size m exceeds 1. While the increase in FLOPs is generally negligible, it becomes more substantial at $m = 28$.

Specifically, with $m = 7$, both $F_\beta^{\max}$ and F_β^w show an improvement of 0.3% over the model without the attention module. These accuracy enhancements are achieved with a slight increase in parameters (0.03M) and FLOPs (0.01G). Among the tested sizes, a mid-size $m = 7$ offers superior performance compared to both larger ($m = 28$) and smaller sizes ($m = 1$ and $m = 3$). Consequently, the proposed GPC with a self-attention module at $m = 7$ effectively enhances estimation accuracy with a negligible efficiency trade-off.

We additionally compare our model with the identical split setting. The results of this comparison are detailed in Table 4. Since both models exhibit equivalent efficiencies, a comparison of efficiency metrics was not conducted. The data demonstrate that accuracy can be slightly enhanced by employing the proposed split ratios, which apply lower dilation ratios to high-scale features.

Table 5 Effect of the global feature extractor (GFE).

Method	#Param (M)	FLOPs (G)	Speed (FPS)	$F_\beta^{\max}$	F_β^w	MAE
w/o GFE	1.61	1.08	699	0.836	0.758	0.051
+ED [18]	1.98	1.21	580	0.853	0.783	0.047
Ours	1.99	1.26	571	**0.867**	**0.804**	**0.042**

Table 6 Ablation study of granularity-aware supervision. F, B, C and O denote the saliency of full map, boundary, center and others, respectively. C-O indicates the combination of center and others foregrounds. Side-outputs are the intermediate representations shown in Fig. 2.

Side-outputs	Setting					
	(a)	(b)	(c)	(d)	(e)	(f) Ours
D_3	F	F	F	F	F	F
D_L	F	B	B	B	-	-
D_H	F	C	O	C-O	-	-
G_f	F	-	-	-	C	-
D_2	F	C-O	C-O	C-O	-	C
D_1	F	B-O	B-O	B-O	B-O	B-O
$F_\beta^{\max}$	0.858	0.848	0.854	0.855	0.854	**0.867**
F_β^w	0.792	0.778	0.786	0.786	0.789	**0.804**
MAE	0.044	0.048	0.046	0.045	0.045	**0.042**

4.3.2 Global feature extractor

Subsequently, we conducted an ablation study to evaluate the impact of the global feature extractor, with the results detailed in Table 5. Our analysis compares our model, which utilizes an efficient attention mechanism for global feature extraction, against two alternatives: one without any global features and another employing extreme downsampling (ED) [18] for global feature extraction. The results indicate that incorporating a global feature extractor significantly enhances estimation accuracy. This improvement corroborates previous findings that global features play a crucial role in salient object detection (SOD) tasks, as highlighted in prior works [18, 49, 51].

Specifically, implementing ED atop the encoder to extract global features notably increases accuracy by 1.7%. The integration of our proposed attention-based feature extractor further amplifies $F_\beta^{\max}$ by an additional 1.4%. Such marked enhancements validate the effectiveness of attention modules in assimilating global features, affirming their utility in complex SOD tasks.

4.3.3 Granularity-aware supervision

Finally, we evaluated the effectiveness of various deep supervision settings within the proposed structure, and the results are summarized in Table 6. Setting (a) serves as the baseline, where both decoder side-outputs and global features are supervised using the full saliency map. In settings (b)-(d), the decoder side-outputs are used to supervise saliencies of different granularity. However, in settings (e) and (f), only the decoder outputs that incorporate global features are utilized for supervision.

The results indicate that supervising side-outputs with different saliency granularities does not generally enhance performance, with the exception of our method. Specifically, the high-level side-output D_2 is supervised using center saliency, and the low-level side-output D_1 is supervised using boundary-other saliency, which leads to improved performance. In contrast, supervising the side-outputs D_L and D_H, which do not integrate global features, does not yield performance gains.

5 Conclusion

In this study, we introduced GAPNet, a lightweight framework for both image and video SOD. With granularity-aware connections, the model fuses low- and high-level features under supervision signals aligned with their granularities, *i.e.*, object locations for coarse levels and boundaries for fine levels. To enhance feature fusion within these connections, we designed granular pyramid convolution with efficient attention (GPC) and cross-scale attention (CSA) strategies tailored to low-level and high-level fusions. Furthermore, a self-attention module was incorporated to capture global information, enabling precise object localization with minimal overhead. Experiments on multiple image and video benchmarks show that GAPNet establishes newstate-of-the-art performance among lightweight models, significantly narrowing the gap to heavyweight counterparts.

Acknowledgements

This work was supported by A*STAR Career Development Fund under grant No. C233312006.

Declarations of Conflict of Interest

The authors declared that they have no conflicts of interest to this work.

References

[1] T. Zhou, D.-P. Fan, M.-M. Cheng, J. Shen, and L. Shao, "Rgb-d salient object detection: A survey," *Computational Visual Media*, vol. 7, pp. 37–69, 2021.

[2] J. Han, D. Zhang, G. Cheng, N. Liu, and D. Xu, "Advanced deep-learning techniques for salient and category-specific object detection: A survey," *IEEE Signal Process. Mag. (SPM)*, vol. 35, no. 1, pp. 84–100, 2018.

[3] W. Wang, J. Shen, X. Dong, and A. Borji, "Salient object detection driven by fixation prediction," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 1711–1720.

[4] C. Liu, Y. Yuan, X. Chen, H. Lu, and D. Wang, "Spatial-temporal initialization dilemma: towards realistic visual tracking," *Visual Intelligence*, vol. 2, no. 1, p. 35, 2024.

[5] Z. Zhang, Z. Cui, C. Xu, Y. Yan, N. Sebe, and J. Yang, "Pattern-affinitive propagation across depth, surface normal and semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 4106–4115.

[6] Y. Liu, Y.-H. Wu, P.-S. Wen, Y.-J. Shi, Y. Qiu, and M.-M. Cheng, "Leveraging instance-, image-and dataset-level information for weakly supervised instance segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.

[7] M.-M. Cheng, F.-L. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu, "Repfinder: finding approximately repeated scene elements for image editing," *ACM Trans. Graphics (TOG)*, vol. 29, no. 4, pp. 1–8, 2010.

[8] Y.-H. Wu, S.-H. Gao, J. Mei, J. Xu, D.-P. Fan, R.-G. Zhang, and M.-M. Cheng, "JCS: An explainable COVID-19 diagnosis

system by joint classification and segmentation," *IEEE Trans. Image Process.*, vol. 30, pp. 3113–3126, 2021.

[9] C. Craye, D. Filliat, and J.-F. Goudou, "Environment exploration for object-based visual saliency learning," in *Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2016, pp. 2303–2309.

[10] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, "Salient object detection: A discriminative regional feature integration approach," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 2083–2090.

[11] W. Wang, Q. Lai, H. Fu, J. Shen, H. Ling, and R. Yang, "Salient object detection in the deep learning era: An in-depth survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 6, pp. 3239–3259, 2021.

[12] N. Liu and J. Han, "DHSNet: Deep hierarchical saliency network for salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 678–686.

[13] P. Zhang, D. Wang, H. Lu, H. Wang, and X. Ruan, "Amulet: Aggregating multi-level convolutional features for salient object detection," in *Int. Conf. Comput. Vis.*, 2017, pp. 202–211.

[14] J.-X. Zhao, J. Liu, D.-P. Fan, Y. Cao, J. Yang, and M.-M. Cheng, "EGNet: Edge guidance network for salient object detection," in *Int. Conf. Comput. Vis.*, 2019, pp. 8779–8788.

[15] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 3917–3926.

[16] Y. Pang, X. Zhao, L. Zhang, and H. Lu, "Multi-scale interactive network for salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 9413–9422.

[17] H. Zhou, X. Xie, J.-H. Lai, Z. Chen, and L. Yang, "Interactive two-stream decoder for accurate and fast saliency detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 9141–9150.

[18] Y.-H. Wu, Y. Liu, L. Zhang, M.-M. Cheng, and B. Ren, "Edn: Salient object detection via extremely-downsampled network," *IEEE Trans. Image Process.*, vol. 31, pp. 3125–3136, 2022.

[19] M. Zhuge, D.-P. Fan, N. Liu, D. Zhang, D. Xu, and L. Shao, "Salient object detection via integrity learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3738–3752, 2022.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Int. Conf. Learn. Represent.*, 2015.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 770–778.

[22] Y.-H. Wu, Y. Liu, X. Zhan, and M.-M. Cheng, "P2T: Pyramid pooling transformer for scene understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 12 760–12 771, 2023.

[23] Y. Liu, Y.-H. Wu, G. Sun, L. Zhang, A. Chhatkuli, and L. Van Gool, "Vision transformers with hierarchical attention," *Machine Intelligence Research*, vol. 21, no. 4, pp. 670–683, 2024.

[24] Y.-H. Wu, S.-C. Zhang, Y. Liu, L. Zhang, X. Zhan, D. Zhou, J. Feng, M.-M. Cheng, and L. Zhen, "Low-resolution self-attention for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2025.

[25] Y. Liu, X.-Y. Zhang, J.-W. Bian, L. Zhang, and M.-M. Cheng, "Samnet: Stereoscopically attentive multi-scale network for lightweight salient object detection," *IEEE Trans. Image Process.*, vol. 30, pp. 3804–3814, 2021.

[26] M.-M. Cheng, J. Warrell, W.-Y. Lin, S. Zheng, V. Vineet, and N. Crook, "Efficient salient region detection with soft image abstraction," in *Int. Conf. Comput. Vis.*, 2013, pp. 1529–1536.

[27] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 4510–4520.

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

[29] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. Torr, "Deeply supervised salient object detection with short connections," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 815–828, 2019.

[30] Y. Liu, Y.-H. Wu, S.-C. Zhang, L. Liu, M. Wu, and M.-M. Cheng, "Revisiting computer-aided tuberculosis diagnosis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 4, pp. 2316–2332, 2024.

[31] G.-P. Ji, G. Xiao, Y.-C. Chou, D.-P. Fan, K. Zhao, G. Chen, and L. Van Gool, "Video polyp segmentation: A deep learning perspective," *Machine Intelligence Research*, vol. 19, no. 6, pp. 531–549, 2022.

[32] G.-P. Ji, J. Liu, P. Xu, N. Barnes, F. S. Khan, S. Khan, and D.-P. Fan, "Frontiers in intelligent colonoscopy," *arXiv preprint arXiv:2410.17241*, 2024.

[33] D.-P. Fan, G.-P. Ji, G. Sun, M.-M. Cheng, J. Shen, and L. Shao, "Camouflaged object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2777–2787.

[34] G.-P. Ji, D.-P. Fan, Y.-C. Chou, D. Dai, A. Liniger, and L. Van Gool, "Deep gradient learning for efficient camouflaged object detection," *Machine Intelligence Research*, vol. 20, no. 1, pp. 92–108, 2023.

[35] Z. Liu, X. Zhang, S. Luo, and O. Le Meur, "Superpixel-based spatiotemporal saliency detection," *IEEE Trans. Circ. Syst. Video Technol. (TCSVT)*, vol. 24, no. 9, pp. 1522–1540, 2014.

[36] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 3166–3173.

[37] X. Huang, Y. Zheng, J. Huang, and Y.-J. Zhang, "50 fps object-level saliency detection via maximally stable region," *IEEE Trans. Image Process.*, vol. 29, pp. 1384–1396, 2019.

[38] X. Li, H. Lu, L. Zhang, X. Ruan, and M.-H. Yang, "Saliency detection via dense and sparse reconstruction," in *Int. Conf. Comput. Vis.*, 2013, pp. 2976–2983.

[39] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 569–582, 2015.

[40] J. Wang, H. Jiang, Z. Yuan, M.-M. Cheng, X. Hu, and N. Zheng, "Salient object detection: A discriminative regional feature integration approach," *Int. J. Comput. Vis.*, vol. 123, no. 2, pp. 251–268, 2017.

[41] Z. Wu, L. Su, and Q. Huang, "Decomposition and completion network for salient object detection," *IEEE Trans. Image Process.*, vol. 30, pp. 6226–6239, 2021.

[42] W. Wang, S. Zhao, J. Shen, S. C. Hoi, and A. Borji, "Salient object detection with pyramid attention and salient edges," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 1448–1457.

[43] W. Wang, J. Shen, M.-M. Cheng, and L. Shao, "An iterative and cooperative top-down and bottom-up inference network for salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 5968–5977.

[44] J. Li, Z. Pan, Q. Liu, and Z. Wang, “Stacked u-shape network with channel-wise attention for salient object detection,” *IEEE Trans. Multimedia*, vol. 23, pp. 1397–1409, 2020.

[45] Z. Yao and L. Wang, “Boundary information progressive guidance network for salient object detection,” *IEEE Trans. Multimedia*, vol. 24, pp. 4236–4249, 2021.

[46] X. Wang, Z. Liu, V. Liesaputra, and Z. Huang, “Feature specific progressive improvement for salient object detection,” *Pattern Recognition*, vol. 147, p. 110085, 2024.

[47] C. Hao, Z. Yu, X. Liu, J. Xu, H. Yue, and J. Yang, “A simple yet effective network based on vision transformer for camouflaged object and salient object detection,” *IEEE Trans. Image Process.*, 2025.

[48] J. Pei, T. Jiang, H. Tang, N. Liu, Y. Jin, D.-P. Fan, and P.-A. Heng, “Calibnet: Dual-branch cross-modal calibration for rgb-d salient instance segmentation,” *IEEE Transactions on Image Processing*, 2024.

[49] S. Chen, X. Tan, B. Wang, and X. Hu, “Reverse attention for salient object detection,” in *Eur. Conf. Comput. Vis.*, 2018, pp. 234–250.

[50] Z. Zhao, C. Xia, C. Xie, and J. Li, “Complementary trilateral decoder for fast and accurate salient object detection,” in *ACM Int. Conf. Multimedia*, 2021, pp. 4967–4975.

[51] Y. K. Yun and W. Lin, “Towards a complete and detail-preserved salient object detection,” *IEEE Trans. Multimedia*, 2023.

[52] J. Li, S. Qiao, Z. Zhao, C. Xie, X. Chen, and C. Xia, “Rethinking lightweight salient object detection via network depth-width tradeoff,” *IEEE Trans. Image Process.*, 2023.

[53] Y. Liu, Y.-C. Gu, X.-Y. Zhang, W. Wang, and M.-M. Cheng, “Lightweight salient object detection via hierarchical visual perception learning,” *IEEE Trans. Cybernetics (TCYB)*, vol. 51, no. 9, pp. 4439–4449, 2021.

[54] C. Fang, H. Tian, D. Zhang, Q. Zhang, J. Han, and J. Han, “Densely nested top-down flows for salient object detection,” *Science China Information Sciences*, vol. 65, no. 8, p. 182103, 2022.

[55] X. Zhou, K. Shen, and Z. Liu, “ADMNet: Attention-guided densely multi-scale network for lightweight salient object detection,” *IEEE Trans. Multimedia*, 2024.

[56] M.-M. Cheng, S.-H. Gao, A. Borji, Y.-Q. Tan, Z. Lin, and M. Wang, “A highly efficient model to study the semantics of salient object detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 8006–8021, 2021.

[57] Z. Wang, Y. Zhang, Y. Liu, D. Zhu, S. A. Coleman, and D. Kerr, “Elwnet: An extremely lightweight approach for real-time salient object detection,” *IEEE Trans. Circ. Syst. Video Technol. (TCSVT)*, 2023.

[58] Z. Wang, Y. Zhang, Y. Liu, C. Qin, S. A. Coleman, and D. Kerr, “Larnet: Towards lightweight, accurate and real-time salient object detection,” *IEEE Trans. Multimedia*, 2023.

[59] Y. Ji, H. Zhang, Z. Zhang, and M. Liu, “Cnn-based encoder-decoder networks for salient object detection: A comprehensive review and recent advances,” *Information Sciences*, vol. 546, pp. 835–857, 2021.

[60] Y.-H. Wu, Y. Liu, L. Zhang, W. Gao, and M.-M. Cheng, “Regularized densely-connected pyramid network for salient instance segmentation,” *IEEE Trans. Image Process.*, vol. 30, pp. 3897–3907, 2021.

[61] J. Chen, H. Zhang, M. Gong, and Z. Gao, “Collaborative compensative transformer network for salient object detection,” *Pattern Recognition*, vol. 154, p. 110600, 2024.

[62] J. Pei, T. Cheng, H. Tang, and C. Chen, "Transformer-based efficient salient instance segmentation networks with orientative query," *IEEE Transactions on Multimedia*, vol. 25, pp. 1964–1978, 2022.

[63] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Int. Conf. Comput. Vis.*, 2015, pp. 1395–1403.

[64] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. Torr, "Deeply supervised salient object detection with short connections," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 5300–5309.

[65] Z. Wu, L. Su, and Q. Huang, "Cascaded partial decoder for fast and accurate salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 3907–3916.

[66] Y. Liu, M.-M. Cheng, X.-Y. Zhang, G.-Y. Nie, and M. Wang, "DNA: Deeply-supervised nonlinear aggregation for salient object detection," *IEEE Trans. Cybernetics (TCYB)*, 2021.

[67] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *Artificial intelligence and statistics.* Pmlr, 2015, pp. 562–570.

[68] J. Wei, S. Wang, Z. Wu, C. Su, Q. Huang, and Q. Tian, "Label decoupling framework for salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 13 025–13 034.

[69] D.-P. Fan, Y. Zhai, A. Borji, J. Yang, and L. Shao, "Bbs-net: Rgb-d salient object detection with a bifurcated backbone strategy network," in *European conference on computer vision.* Springer, 2020, pp. 275–292.

[70] Z. Luo, N. Liu, W. Zhao, X. Yang, D. Zhang, D.-P. Fan, F. Khan, and J. Han, "Vscode: General visual salient and camouflaged object detection with 2d prompt learning," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024, pp. 17 169–17 180.

[71] B.-W. Yin and Z. Lin, "Exploring salient object detection with adder neural networks," in *AAAI Conf. Artif. Intell.*, vol. 39, no. 9, 2025, pp. 9490–9498.

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

[73] B. Wang, W. Liu, G. Han, and S. He, "Learning long-term structural dependencies for video salient object detection," *IEEE Trans. Image Process.*, vol. 29, pp. 9017–9031, 2020.

[74] C. Chen, G. Wang, C. Peng, Y. Fang, D. Zhang, and H. Qin, "Exploring rich and efficient spatial temporal interactions for real-time video salient object detection," *IEEE Trans. Image Process.*, vol. 30, pp. 3995–4007, 2021.

[75] G.-P. Ji, K. Fu, Z. Wu, D.-P. Fan, J. Shen, and L. Shao, "Full-duplex strategy for video object segmentation," in *Int. Conf. Comput. Vis.*, 2021, pp. 4922–4933.

[76] S. Gao, H. Xing, W. Zhang, Y. Wang, Q. Guo, and W. Zhang, "Weakly supervised video salient object detection via point supervision," in *ACM Int. Conf. Multimedia*, 2022, pp. 3656–3665.

[77] R. Cong, W. Song, J. Lei, G. Yue, Y. Zhao, and S. Kwong, "Psnet: Parallel symmetric network for video salient object detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 7, no. 2, pp. 402–414, 2022.

[78] R. Qian, W. Lin, J. See, and D. Li, "Controllable augmentations for video representation learning," *Visual Intelligence*, vol. 2, no. 1, p. 1, 2024.

[79] B. Chen, Z. Chen, X. Hu, J. Xu, H. Xie, J. Qin, and M. Wei, "Dynamic message propagation network for rgb-d and video

salient object detection," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 20, no. 1, pp. 1–21, 2023.

[80] S. Ren, C. Han, X. Yang, G. Han, and S. He, "Tenet: Triple excitation network for video salient object detection," in *Eur. Conf. Comput. Vis.* Springer, 2020, pp. 212–228.

[81] M. Zhang, J. Liu, Y. Wang, Y. Piao, S. Yao, W. Ji, J. Li, H. Lu, and Z. Luo, "Dynamic context-sensitive filtering network for video salient object detection," in *Int. Conf. Comput. Vis.*, 2021, pp. 1553–1563.

[82] X. Zhao, H. Liang, P. Li, G. Sun, D. Zhao, R. Liang, and X. He, "Motion-aware memory network for fast video salient object detection," *IEEE Trans. Image Process.*, vol. 33, pp. 709–721, 2024.

[83] N. Liu, K. Nan, W. Zhao, X. Yao, and J. Han, "Learning complementary spatial–temporal transformer for video salient object detection," *IEEE Trans. Neur. Net. Learn. Syst.*, vol. 35, no. 8, pp. 10 663–10 673, 2024.

[84] Y.-X. Li, C.-L.-Z. Chen, S. Li, A.-M. Hao, and H. Qin, "A novel divide and conquer solution for long-term video salient object detection," *Machine Intelligence Research*, vol. 21, no. 4, pp. 684–703, 2024.

[85] N. Liu, J. Han, and M.-H. Yang, "Picanet: Pixel-wise contextual attention learning for accurate saliency detection," *IEEE Trans. Image Process.*, vol. 29, pp. 6438–6451, 2020.

[86] Y.-H. Wu, Y. Liu, J. Xu, J.-W. Bian, Y.-C. Gu, and M.-M. Cheng, "MobileSal: Extremely efficient rgb-d salient object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021.

[87] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *International Conference on 3D Vision*. IEEE, 2016, pp. 565–571.

[88] N. Liu, N. Zhang, K. Wan, L. Shao, and J. Han, "Visual saliency transformer," in *Int. Conf. Comput. Vis.*, 2021, pp. 4722–4732.

[89] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Adv. Neural Inform. Process. Syst.*, 2019, pp. 8026–8037.

[90] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Int. Conf. Learn. Represent.*, 2015.

[91] K. Fu, D.-P. Fan, G.-P. Ji, Q. Zhao, J. Shen, and C. Zhu, "Siamese network for rgb-d salient object detection and beyond," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5541–5559, 2022.

[92] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 2462–2470.

[93] L. Wang, H. Lu, Y. Wang, M. Feng, D. Wang, B. Yin, and X. Ruan, "Learning to detect salient objects with image-level supervision," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 136–145.

[94] G. Li and Y. Yu, "Visual saliency based on multiscale deep features," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 5455–5463.

[95] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 1155–1162.

[96] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille, "The secrets of salient object segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 280–287.

[97] T. Wang, L. Zhang, S. Wang, H. Lu, G. Yang, X. Ruan, and A. Borji, "Detect globally, refine locally: A novel approach to saliency detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 3127–3135.

[98] Y. Zeng, H. Lu, L. Zhang, M. Feng, and A. Borji, "Learning to promote saliency detectors," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 1644–1653.

[99] Y. Wang, R. Wang, X. Fan, T. Wang, and X. He, "Pixels, regions, and objects: Multiple enhancement for salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023, pp. 10 031–10 040.

[100] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 724–732.

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

[102] 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, 2015.

[103] R. Margolin, L. Zelnik-Manor, and A. Tal, "How to evaluate foreground maps?" in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 248–255.

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

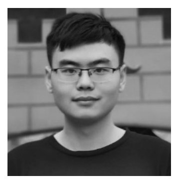
[105] D.-P. Fan, C. Gong, Y. Cao, B. Ren, M.-M. Cheng, and A. Borji, "Enhanced-alignment measure for binary foreground map evaluation," in *IJCAI*, 2018, pp. 698–704.

[106] J. Zhao, Y. Zhao, J. Li, and X. Chen, "Is depth really necessary for salient object detection?" in *ACM Int. Conf. Multimedia*, 2020, pp. 1745–1754.

[107] H. Li, G. Chen, G. Li, and Y. Yu, "Motion guided attention for video salient object detection," in *Int. Conf. Comput. Vis.*, 2019, pp. 7274–7283.

[108] P. Yan, G. Li, Y. Xie, Z. Li, C. Wang, T. Chen, and L. Lin, "Semi-supervised video salient object detection using pseudo-labels," in *Int. Conf. Comput. Vis.*, 2019, pp. 7284–7293.

[109] W. Zhao, J. Zhang, L. Li, N. Barnes, N. Liu, and J. Han, "Weakly supervised video salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 16 826–16 835.



Yu-Huan Wu received his Ph.D. degree from Nankai University in 2022. He is a research scientist at the Institute of High Performance Computing (IHPC), A*STAR, Singapore. He has published 10+ papers on top-tier conferences and journals such as IEEE TPAMI/TIP/TNNLS/CVPR/ICCV. His research interests include computer vision and deep learning.

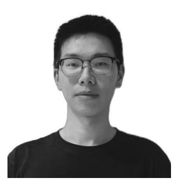
E-mail: wyh.nku@gmail.com

ORCID iD: 0000-0001-8666-3435

Wei Liu received the bachelor's and master's degrees from Huazhong University of Science and Technology, China in 2015 and 2018, respectively. He then obtained the Ph.D. degree from Nanyang Technological University, Singapore in 2022. He is currently a research scientist at Institute of High Performance Computing (IHPC), Agency for Science, Technology and Research (A*STAR), Singapore. His research interests include computer vision and efficient machine learning.

E-mail: liuw1204@gmail.com (Corresponding author)

ORCID iD: 0000-0002-9770-8923



Zi-Xuan Zhu received his bachelor's degree from Nankai University in 2025. He is pursuing his doctoral degree under the supervision of Prof. Deng-Ping Fan in Nankai University. His research interests include computer vision and deep learning.

E-mail: zzxnku@mail.nankai.edu.cn

ORCID iD: 0009-0006-4357-4233

Zizhou Wang received his Ph.D. degree in computer science from Sichuan University in 2022. He is currently a research scientist in Institute of High Performance Computing (IHPC), Agency for Science, Technology and Research (A*STAR), Singapore. His current research interests include robust machine learning and Intelligent medical imaging.

E-mail: wang_zizhou@ihpc.a-star.edu.sg

ORCID iD: 0000-0003-2234-9409

Yong Liu is Deputy Department Director, Computing & Intelligence Department at Institute of High Performance Computing (IHPC), A*STAR, Singapore. He is also Adjunct Associate Professor at Duke-NUS Medical School, NUS and Adjunct Principal Investigator at Singapore Eye Research Institute (SERI). He has led multiple research projects in multimodal machine learning, medical imaging analysis, especially AI in healthcare.

E-mail: liuyong@ihpc.a-star.edu.sg

ORCID iD: 0000-0002-1590-2029

Liangli Zhen received his Ph.D. degree from Sichuan University in 2018. He is a senior scientist and group manager at the Institute of High Performance Computing (IHPC), A*STAR, Singapore. His research interests include machine learning and optimization. He has led/co-led multiple research initiatives in robust multimodal learning. His research findings have been published in top tier journals and conferences, including IEEE TPAMI, TNNLS, ICCV, and CVPR.

E-mail: llzhen@outlook.com (Corresponding author)

ORCID iD: 0000-0003-0481-3298