Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.060653

ARTICLE

Efficient Spatiotemporal Information Utilization for Video Camouflaged Object Detection

Dongdong Zhang, Chunping Wang, Huiying Wang and Qiang Fu*

Army Engineering University of PLA, Shijiazhuang, 050003, China

*Corresponding Author: Qiang Fu. Email: Fu_Qiang@aeu.edu.cn

Received: 06 November 2024; Accepted: 20 December 2024; Published: 06 March 2025

ABSTRACT: Video camouflaged object detection (VCOD) has become a fundamental task in computer vision that has attracted significant attention in recent years. Unlike image camouflaged object detection (ICOD), VCOD not only requires spatial cues but also needs motion cues. Thus, effectively utilizing spatiotemporal information is crucial for generating accurate segmentation results. Current VCOD methods, which typically focus on exploring motion representation, often ineffectively integrate spatial and motion features, leading to poor performance in diverse scenarios. To address these issues, we design a novel spatiotemporal network with an encoder-decoder structure. During the encoding stage, an adjacent space-time memory module (ASTM) is employed to extract high-level temporal features (i.e., motion cues) from the current frame and its adjacent frames. In the decoding stage, a selective space-time aggregation module is introduced to efficiently integrate spatial and temporal features. Additionally, a multi-feature fusion module is developed to progressively refine the rough prediction by utilizing the information provided by multiple types of features. Furthermore, we incorporate multi-task learning into the proposed network to obtain more accurate predictions. Experimental results show that the proposed method outperforms existing cutting-edge baselines on VCOD benchmarks.

KEYWORDS: Video camouflaged object detection; spatiotemporal information; feature fusion; multi-task learning

1 Introduction

Camouflaged object detection (COD) aims to detect and segment objects that are hidden in their surroundings. These objects closely resemble the background in texture, color, and shape, and their boundaries are often ambiguous. As a result, COD is more challenging than other tasks, such as general object detection [1] and salient object detection [2]. Recently, COD has attracted interest from many researchers and facilitated various practical applications, including agricultural management, industrial defect detection, and medical analysis (e.g., polyp segmentation, lung infection segmentation). In terms of data types, the study of COD can be categorized into two groups: image-based COD (ICOD), and video-based COD (VCOD). While the former detects camouflaged objects in a single image by mining semantic information in the space, the latter attempts to identify camouflaged objects by extracting motion cues from consecutive video frames.

In recent years, with the development of deep learning and the proposal of high-quality pixel-wise annotated large-scale benchmark datasets, ICOD has been extensively researched. Numerous methods based on Convolutional Neural Networks (CNNs) [3–5] have been developed, achieving notable progress. However, due to the strong visual resemblances between camouflaged objects and their surroundings in terms of texture, color, or boundaries, it is difficult to excavate discriminative semantic cues for camouflaged

objects from the appearance and geometric information provided by static images, resulting in the existing COD methods still struggling to accurately and reliably segment the camouflaged objects from the chaotic backgrounds. To address these challenges, recent research [6–9] has explored the incorporation of temporal motion cues, aiming to enhance COD performance by expanding the information dimensions. However, processing motion cues presents challenges due to complex motion patterns, variable visual appearance, and cluttered backgrounds.

Current research on VCOD primarily focuses on leveraging or modeling motion cues. One study [6] employs optical flow generated by existing motion estimation models to provide motion cues for identifying camouflaged objects. Notably, even state-of-the-art (SOTA) optical flow models fail to estimate motions for camouflaged objects. The error in optical flow estimation accumulates with the duration of the video and eventually affects the segmentation performance of the network. To address this issue, another study [7] develops a short-term dynamic module to implicitly capture the motion between consecutive frames. However, the reliability of motion learning cannot be assured due to the absence of explicit regularization or evaluation in implicit modeling. To tackle these challenges, a novel VCOD framework of Explicit Motion handling and Interactive Prompting is proposed in [8]. This framework explicitly handles motion cues using an optical flow model and supervises the optical flow to learn the prompts fed to the motion flow in a self-supervised manner. Consequently, the framework not only ensures the reliability of motion learning, but also avoids the accumulation of optical flow errors, which improves the accuracy of segmentation. Although the above methods have achieved success in extracting motion cues, motion cues are unstable. For instance, if an object remains stationary for an extended period, the motion cues may be entirely absent [9]. In addition, due to the concealed nature of camouflaged objects, motion cues can only provide an indication of rough object motion, but not fine-grained details, e.g., the exact shape of the objects and their contours. Inspired by ICOD works, spatial information should be emphasized in VCOD tasks, as it tends to be more stable than motion cues for video data. Therefore, we believe that beyond motion cues, spatial information is also crucial for VCOD tasks. However, efficiently combining motion cues and spatial information to achieve fine-grained segmentation of camouflaged objects in space is an urgent research problem that needs to be addressed.

Space-time Memory Network (STM) has demonstrated its effectiveness in capturing temporal information for video object segmentation (VOS) and video saliency object detection (VSOD) tasks [10,11]. Therefore, we employ the Adjacent Space-time Memory Module (ASTM) [11] as the temporal branch of our model to collect temporal information from adjacent frames, i.e., to obtain motion cues from adjacent frames. In the decoding stage, to efficiently combine temporal and spatial information, we propose a selective space-time aggregation module (SSAM). This module is designed to facilitate collaborative learning between temporal and spatial features and the biased combination of these features based on the object state (motion or stationary) to improve the efficiency of information utilization. Inspired by the stepwise refinement strategy adopted in ICOD, we develop a multi-feature fusion module (MFFM) to refine segmentation results by focusing on ambiguous regions. Specifically, this module progressively integrates high-level features and predictions to refine features from coarse to fine, generating more accurate predictions. Furthermore, to obtain more accurate predictions, we incorporate the boundary detection task into our network and construct a multi-task detection head. During multi-task processing, we enhance the interaction between the boundary detection task and the segmentation task to fully exploit the mutually beneficial information between them. The main contributions of this paper can be summarized as follows:

(1) We incorporate the STM mechanism into VCOD as a temporal branch to capture motion cues from adjacent frames, providing an alternative for exploiting or modeling motion cues in VCOD tasks. We propose an efficient space-time fusion module for mutual modulation and adaptive selective fusion between temporal and spatial information.

(2) We design a multi-feature fusion module to generate fine-grained prediction maps by leveraging the complementary features of multiple types. In a coarse-to-fine manner, the module can accurately locate objects and substantially eliminate ambiguous regions in space.

(3) We introduce boundary prediction into VSOD and enhance the interaction between different tasks, enabling the model to focus on the perception of object boundaries while concentrating on object integrity.

(4) We evaluate the proposed method on two widely recognized VCOD benchmarks, and the experimental results demonstrate that our method obtains better segmentation results than SOTA methods.

2 Related Work

In this section, we first provide a concise overview of ICOD and VCOD models proposed in recent years, followed by an introduction to memory networks and their variants. In addition, we present the motivation for developing our model inspired by related work.

ICOD. ICOD methods aim to discern camouflaged objects from a single RGB image. Early COD methods relied on various handcrafted features (e.g., color, texture, and gradient) to distinguish camouflaged objects from the background [12–14]. However, due to the limited expressiveness of handcrafted features, early methods are only effective in relatively simple scenes and often suffer from significant performance degradation or even failure when faced with complex scenes or when the camouflaged object features closely resemble the background. With the development of deep learning and the establishment of large-scale image COD datasets (CAMO [15], COD10K [4] and NC4K [16]), COD methods have undergone substantial progress in recent years. Inspired by natural predatory behaviors or human visual mechanisms, some works [3,4,17,18] employed a stepwise refinement strategy, first making a rough prediction of camouflaged regions and then refining the prediction by various measures. Among them, Zhuge et al. [18] refined the segmentation results by constructing an efficient feature fusion strategy. However, without additional cues, the model is difficult to distinguish the part of the region where camouflaged objects are extremely similar to the background by the feature fusion strategy alone. To improve segmentation performance, some studies [15,16,19,20] constructed multi-task learning frameworks and introduced tasks like classification, ranking, and gradient estimation for auxiliary the main segmentation task to obtain satisfactory results. Zhong et al. [20] attempted to address the intrinsic similarity between foreground and background for COD in the frequency domain. To obtain fine boundary contours, references [21–24] adopt boundary cues as a focus of model learning. Specifically, Qin et al. [21] designed a hybrid loss to implicitly direct the network to pay more attention to object boundary information, which is able to obtain fine camouflaged object boundaries without explicitly extracting the boundaries. Ji et al. [22] and Zhang et al. [23] obtained boundary priors through displaying supervision and refined the object by fusing multiple information. Zhai et al. [24] treated boundary detection as a parallel task to segmentation task and detected camouflaged objects by reasoning about the complementary information of the two tasks.

VCOD. Motion helps reveal camouflaged objects in video data by distinguishing them from their surroundings. While Lamdouar et al. [6] proposed frame alignment and motion segmentation using difference images and optical flow, even state-of-the-art optical flow algorithms struggle with camouflaged objects, leading to cumulative errors. To address this, Cheng et al. [7] developed a dense volume approach with two-stage refinement, while Zhang et al. [8] created a two-stream architecture combining flow estimation and segmentation. Our approach differs by learning temporal information directly from frames without optical flow estimation. Given the similarity between video and image camouflaged object detection, we incorporate successful strategies from image-based methods, including boundary detection and multi-feature fusion for progressive refinement of segmentation results.

Memory Network. The memory network was first proposed for natural language processing, memorizing external information as key and value, and generating a correlation graph through non-local matching between query and key. As an alternative to Long Short-Term Memory, memory networks have been widely applied in a variety of computer vision tasks [25–27]. Oh et al. [10] applied memory networks to VOS and proposed a novel space-time memory network (STM), achieving SOTA results at that time and promoting the development of other STM variants in VOS. Cheng et al. [28] simplified the STM to construct a minimalist form of a matching network. Liang et al. [29] designed a dynamic memory network to obtain the current frame representation of target objects from the visual content of all past frames. Shaker et al. [30] introduced an optimized dynamic long-term modulated cross-attentive memory to encode only target-relevant information, substantially reducing computational complexity. Recognizing the successful application of STM in VOS and its powerful ability to capture temporal information, Zhao et al. [11] applied STM to VSOD and designed an ASTM built upon high-level features. The above work mainly focuses on efficiently modeling temporal information and does not consider how to efficiently fuse temporal and spatial information after obtaining them. For VCOD, we aim for temporal and spatial features to collaborate with each other in learning and to fuse temporal and spatial information in a biased way according to the state (motion or stationary) of objects. When the object is in motion, we want the network to be biased towards temporal information, and when the object is in a stationary state, we want the network to be biased towards spatial information. Considering the similarity between VCOD and VSOD, we directly employ ASTM to extract temporal information.

3 Method

In this section, we first present an overview of the network architecture. Then, we describe the key module, selective space-time aggregation module (SSAM), for efficiently integrating temporal and spatial information. Subsequently, we introduce the multi-feature fusion module (MFFM) for efficiently fusing various types of features. Furthermore, we construct a multi-task detection head with a boundary detection task for applying the multi-task learning framework to VCOD. Finally, we provide a detailed explanation of the training loss for the entire model.

3.1 Overall Architecture

As shown in Fig. 1, our model adopts an encoder-decoder structure. Given a consecutive frame sequence X = $(x_1, x_2, \ldots, x_T)$, the current frame for processing is denoted as x_t. The entire encoding phase follows [11], employing two ResNet [31] with shared weights as parallel encoders, one to memorize the temporal information of the previous frame x_{t-1} and the next frame x_{t+1}, denoted as E_M, and the other to obtain the spatial information of the current frame, denoted as E_Q. The high-level features E_Q^{res5} of current frame and high-level features E_M^{res5} of adjacent frames are fed into ASTM for generating temporal high-level features E_t. During the decoding process, temporal high-level features E_t and spatial high-level features E_Q^{res5} of current frame are fed into the SSAM. This facilitates learning between temporal and spatial features and adaptively updates the weight assignments based on the state of the current object, enhancing the utilization of temporal and spatial information. The detection head employed in this paper, which consists of a segmentation branch and a boundary detection branch, is specially designed to facilitate the propagation of useful information between the segmentation task and the boundary detection task. Finally, a MFFM is exploited to refine the object regions and boundaries, and accurate detection results will be generated by the detection head. Detailed procedure of decoding is described in Algorithm 1. Different from previous approaches [6,8] that rely on optical flow to establish temporal information, our network does not require external temporal information and can be trained in an end-to-end manner. The design and implementation

details of sub-modules in our proposed network are described in the following subsections. It should be noted that ASTM, which is not our focus, will not be discussed in detail, and further information can be found in the literature [11].

Algorithm 1: Decoding process

Input: Multi-level features from encoder: $\left\{E_Q^{resi}, i \in \{2,3,4,5\}\right\}$ Temporal features: E_t

Output: Mask prediction: $\left\{m_t^i, i \in \{2,3,4,5\}\right\}$ Boundary prediction: $\left\{e_t^i, i \in \{2,3,4,5\}\right\}$

for $i = 5; i >= 2; i-$ do
  if $i == 5$ **then**
    $\widetilde{E}_t = SSAM\left(E_Q^{res5}, E_t\right)$
    $m_t^5, e_t^5, f_m^5, f_e^5 = Head\left(\widetilde{E}_t\right)$
  else if $i == 4$ **then**
    $f_s^4 = MFFM\left(E_Q^{res4}, \widetilde{E}_t, f_m^5, f_e^5\right)$
    $m_t^4, e_t^4, f_m^4, f_e^4 = Head\left(f_s^4\right)$
  else
    $f_s^i = MFFM\left(E_Q^{resi}, f_s^{i+1}, f_m^{i+1}, f_e^{i+1}\right)$
    $m_t^i, e_t^i, f_m^i, f_e^i = Head\left(f_s^i\right)$
end for
return Mask prediction: $\left\{m_t^i, i \in \{2,3,4,5\}\right\}$ Edge prediction: $\left\{e_t^i, i \in \{2,3,4,5\}\right\}$

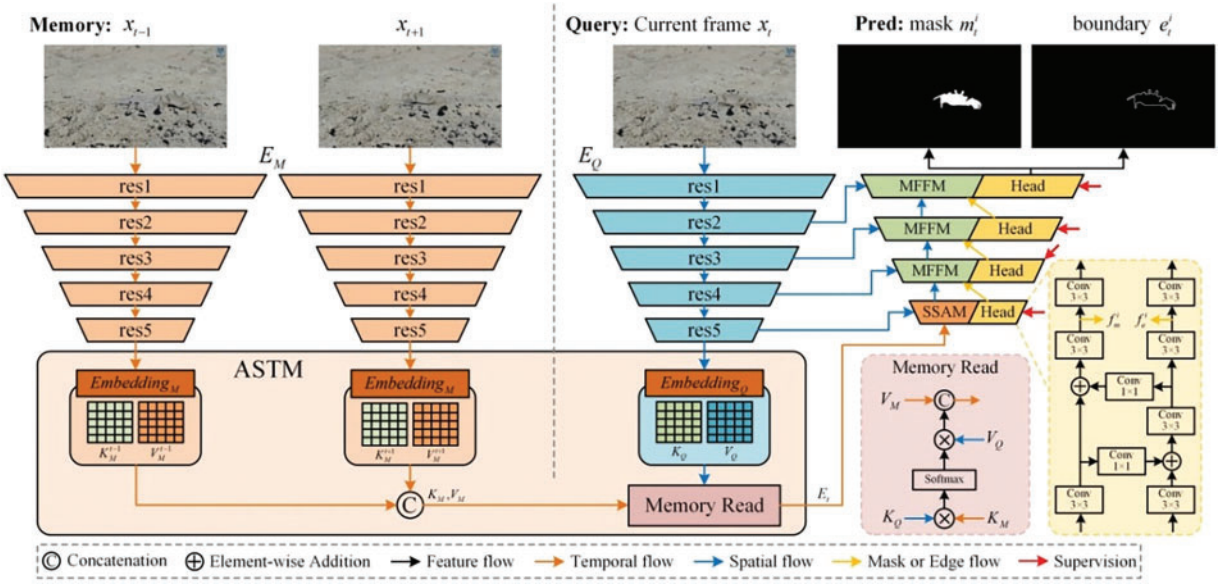


Figure 1: Overall structure of the model

3.2 Selective Space-Time Aggregation Module

For VCOD tasks, motion cues play a crucial role in breaking camouflage, as camouflaged objects often become salient and easily detectable when in motion. However, the motion of camouflaged objects is complex and variable, alternating between motion and stationary states. When an object is stationary, the motion cues will disappear, and only spatial cues can be utilized to identify the camouflaged object. In the process of fusing

temporal and spatial information, directly treating temporal and spatial information as equally important without considering the motion state of the object may lead to information waste. Moreover, temporal and spatial features should not be treated independently.

To address the aforementioned problems, we propose the SSAM for enhancing learning between temporal and spatial features and performing adaptive bias aggregation of temporal and spatial information. As shown in Fig. 2, SSAM takes temporal features E_t and spatial features E_Q^{res5} as inputs and outputs spatiotemporal features $\widetilde{E}$. First, we transform E_t and E_Q^{res5} into query (Q_t, Q_r), key (K_t, K_r), and value (V_t, V_r) via MLP layers, respectively. Attention is employed for cross-domain modulation between temporal and motion information, computed by extracting the key from one domain and the query from another domain, and generated via Softmax. Then, we apply the attention to the query in the current domain and residually connect it with the original features, which in turn yields enhanced features $\widetilde{E}_t$ and $\widetilde{E}_Q^{res5}$. The calculation process can be expressed as follows:

$$\begin{aligned}\widetilde{E}_t &= Softmax\left(Q_r \times K_t/\sqrt{d}\right) \times V_t + E_t \\ \widetilde{E}_Q^{res5} &= Softmax\left(Q_t \times K_r/\sqrt{d}\right) \times V_r + E_Q^{res5}\end{aligned} \tag{1}$$

where d denotes the dimension of linear projection, $\times$ and $+$ denote element-wise multiplication and addition operation, respectively. After temporal-enhanced features $\widetilde{E}_t$ and spatial-enhanced features $\widetilde{E}_Q^{res5}$ are obtained by cross-domain modulation, selective bias mechanism is used to adaptively learn a weight for weighing temporal and spatial features. The weight is generated as:

$$w = \sigma\left(FC\left(Cat\left(GAP\left(\widetilde{E}_t\right), GAP\left(\widetilde{E}_Q^{res5}\right)\right)\right)\right) \tag{2}$$

where σ is sigmoid function, FC is a full connectivity layer, Cat is channel-wise concatenation operation, and GAP is global average pooling.

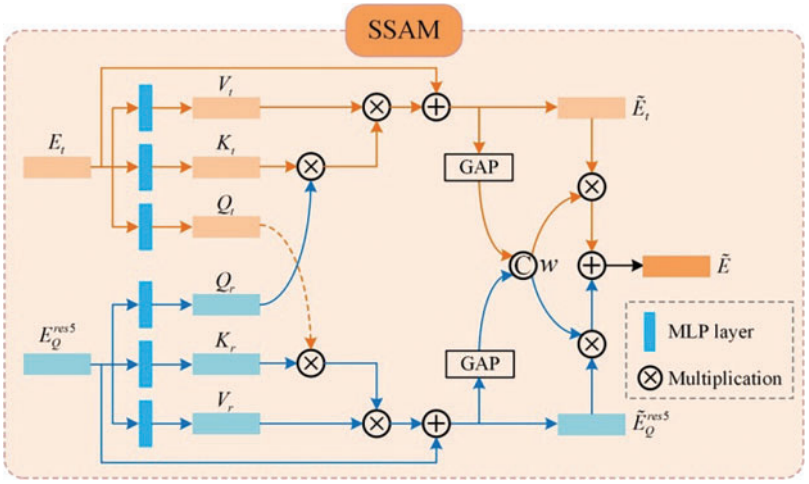


Figure 2: Detailed structure of the SSAM

Finally, the spatiotemporal features $\widetilde{E}$ are obtained by fusing the temporal-enhanced features and spatial-enhanced features under the modulation of weights. The fusion process can be formulated as:

$$\widetilde{E} = w \odot \widetilde{E}_t + (1 - w) \odot \widetilde{E}_Q^{res5} \tag{3}$$

where $\odot$ denotes element-wise multiplication with the broadcasting strategy.

3.3 Multi-Feature Fusion Module

It is well known that high-level features contain semantic information for localizing objects, while low-level features tend to retain spatial details for constructing object boundaries. With high-level spatiotemporal features, we can only locate the rough position of camouflaged objects, and need to utilize low-level spatial features to further refine the camouflaged regions and boundaries. Due to the extremely high intrinsic similarity between camouflaged objects and their surroundings, it is difficult to locate camouflaged objects in low-level features with rich detail information without additional cues for guidance. Therefore, we propose a multi-feature fusion module to enable the network to be more focused on ambiguous regions by fusing high-level features and rough predictions. Furthermore, MFFM is applied to three layers of low-level features (res4, res3, and res2) to realize the gradual refinement of camouflaged regions. Compared with some existing fusion modules [18,22–24], MFFM is able to effectively fuse multiple types of features, and its structure is shown in Fig. 3. First, we fuse the low-level features E_Q^{resi} with the high-level features f_s^{i+1} or spatiotemporal features $\widetilde{E}$ to enrich spatial semantic information. Then, we perform a reverse operation on the rough prediction map f_m^{i+1} to obtain the reverse attentional guidance. Subsequently, we perform element-wise multiplication and jump connection between the fused features and the attentional guidance to obtain the initial fused features $\widehat{E}_Q^{resi}$, which can be denoted as:

$$\begin{aligned} \widehat{E}_Q^{resi} &= E_Q^{resi} + f_s^{i+1} \\ \widehat{E}_Q^{resi} &= f^{3\times3}\left(\widehat{E}_Q^{resi} \times \left(1 - \sigma\left(f_m^{i+1}\right)\right) + \widehat{E}_Q^{resi}\right) \end{aligned} \tag{4}$$

where $f^{3\times3}$ is 3×3 convolution. To exclude unreasonable regions and enhance the feature representation of object regions, we introduce local channel attention to highlight object-related feature channels. Specifically, global average pooling is first adopted to aggregate features. Then, the weights of each channel are obtained via a 1×1 convolution and a subsequent sigomid function. Finally, the feature $\widetilde{f}_s^i$ will be obtained by multiplying the channel weights with the initial fusion feature $\widehat{E}_Q^{resi}$ and using 1×1 convolution to reduce the number of channels. The process can be depicted as:

$$\widetilde{f}_s^i = f^{1\times1}\left(\sigma\left(f_k^{1\times1}\left(GAP\left(\widehat{E}_Q^{resi}\right)\right)\right) \times \widehat{E}_Q^{resi}\right) \tag{5}$$

where $f^{1\times1}$ is 1×1 convolution, $f_k^{1\times1}$ is 1×1 convolution with kernel size k. The convolution kernel size k can be set autonomously according to the number of channels C of $\widehat{E}_Q^{resi}$, which can be calculated as:

$$k = \left|\left(1 + \log_2 C\right)/2\right| \tag{6}$$

where $|\cdot|$ denotes the rounding down operation. Obviously, utilizing channel attention strategy can suppress redundant channels and highlight critical channels, which can filter out unrelated regions and enable the network to pay more attention to the regions where camouflaged objects exist.

Finally, $\widetilde{f}_s^i$ is cascaded with boundary features f_e^{i+1} to enhance the feature representation. The final output f_s^i can be represented as:

$$f_s^i = Cat\left(\widetilde{f}_s^i, f_e^{i+1}\right) \tag{7}$$

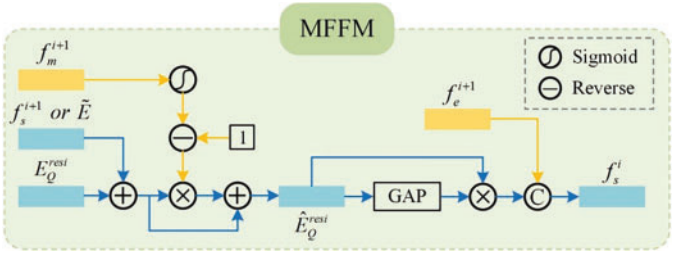


Figure 3: The architecture of our proposed MFFM

3.4 Multi-Task Detection Head

In the ICOD task, several previous studies [21–24] have demonstrated that the introduction of a multi-task learning framework can facilitate the learning of segmentation and boundary detection tasks, which mutually enhances the performance of each task. Inspired by the instance segmentation work [32], we add a boundary detection branch parallel to the segmentation branch in the detection head, forming a multi-task detection head that makes the model focus more on object boundaries and shapes. Intuitively, object masks and object boundaries have a close relationship, i.e., mask features can provide semantic information for learning boundaries, and shape and location information in boundary features can guide more precise mask prediction. Furthermore, object masks and object boundaries can be easily converted to each other. Therefore, there is a large amount of mutually exploitable information between the segmentation branch and the boundary detection branch. To facilitate their interaction, we adopt a fusion strategy [32] to fully leverage the special relationship between mask features and boundary features. The structure of the multi-task detection head is shown in the lower right corner of Fig. 1. First, we perform a 3 × 3 convolution on spatiotemporal feature $\widetilde{E}$ or feature f_s^i to obtain the initial mask feature $\check{f}_m^i$ and boundary feature $\check{f}_e^i$. Then, $\check{f}_e^i$ is fused with $\check{f}_m^i$ operated by a 1 × 1 convolution operation and a 3 × 3 convolution is applied to obtain the fusion feature $\widehat{f}_e^i$. Subsequently, $\check{f}_m^i$ is fused with $\widehat{f}_e^i$ operated by a 1 × 1 convolution operation to obtain the fusion feature $\widehat{f}_m^i$. Finally, two 3 × 3 convolution operations are applied to $\widehat{f}_m^i$ and $\widehat{f}_e^i$, respectively, to obtain the final mask prediction map m_t^i and boundary prediction map e_t^i. The above process can be formulated as follows:

$$\begin{aligned}
\check{f}_e^i &= f^{3\times3}\left(f_s^i\right) \\
\check{f}_m^i &= f^{3\times3}\left(f_s^i\right) \\
\widehat{f}_e^i &= f^{3\times3}\left(f^{1\times1}\left(\check{f}_m^i\right) + \check{f}_e^i\right) \\
\widehat{f}_m^i &= \check{f}_m^i + \widehat{f}_e^i \\
m_t^i &= f^{3\times3}\left(f^{3\times3}\left(\widehat{f}_m^i\right)\right) \\
e_t^i &= f^{3\times3}\left(f^{3\times3}\left(\widehat{f}_e^i\right)\right)
\end{aligned} \tag{8}$$

3.5 Loss Function

Our network is a multi-task framework that contains two tasks: segmentation and edge detection. For segmentation task, following previous work [7,8], we adopt a hybrid loss function, which includes weighted cross-entropy loss L_{ce}^w, weighted intersection-over-union loss L_{iou}^w, and enhanced-alignment loss L_e. The hybrid loss L_{seg} can be defined as:

$$L_{seg} = L_{ce}^w + L_{iou}^w + L_e \tag{9}$$

For boundary detection task, following previous work [32], we use dice loss L_{dic} and weighted cross-entropy loss L_{ce}^{w} to optimize the boundary prediction. The boundary loss L_{bou} can be expressed as follows:

$$L_{bou} = L_{dic} + L_{ce}^{w} \tag{10}$$

As shown in Fig. 1, we perform multiple supervisions for the mask prediction maps and boundary prediction maps output from four layers. Here, each prediction map is upsampled to have the same size as the ground truth map. Therefore, the overall loss L_{total} can be formulated as follows:

$$L_{total} = \sum_{i=2}^{5} L_{seg}\left(m_t^i, m\right) + \sum_{i=2}^{5} L_{bou}\left(e_t^i, e\right) \tag{11}$$

where m and e denote mask and boundary ground-truth maps, respectively.

4 Experiments

In this section, we initially provide an introduction to the dataset utilized in our experiments, the evaluation metrics employed, and the details of model training. Subsequently, a comprehensive evaluation of our model is performed in both quantitative and qualitative terms. Furthermore, ablation studies are conducted to validate the effectiveness of our proposed components.

4.1 Experimental Setup

1) Datasets: we utilize the most widely used datasets in the COD field for model training and performance evaluation, including COD10K [3], MoCA-Mask [33], and CAD [34].

COD10K. COD10K is currently the largest camouflaged object dataset with high-quality pixel-level annotations, containing 5066 camouflaged images, of which 3040 are used for training and 2026 for testing.

MoCA-Mask. MoCA-Mask is reorganized from the MoCA dataset and annotated with segmentation. The dataset contains 87 sequences, of which 71 sequences with 19,313 frames are for training and the remaining 16 sequences with 3626 frames are for testing. These images are sampled from YouTube videos and have a resolution of 720 × 1280 in most cases.

CAD. CAD is a small VCOD dataset containing 9 short video sequences with accompanying hand-labeled ground truth every 5th frame.

In our experiments, the COD10K dataset is employed to pre-train all ICOD methods as well as encoders of VCOD methods. Except for the 71 sequences in the MoCA-Mask training set that are employed for training, the rest of the video sequences are used for testing [7].

2) Evaluation Metrics: We adopt the same evaluation metrics used in [7] to assess model performance, i.e., Mean Absolute Error (M), Enhanced-alignment Measure (E_ϕ), Weighted F-measure (F_β^ω), Structure Measure (S_α), mean Dice (mDic), and mean IoU (mIoU). Evaluation code: https://github.com/lartpang/PySODEvalToolkit (accessed on 30 November 2024).

3) Training Details: As demonstrated in previous work [7], pre-training the model using the COD10K dataset can further improve its performance on MoCA-Mask dataset. Following [7], we pre-trained the proposed model on the COD10K dataset and fine-tuned it on the MoCA-Mask training set. Specifically, we first perform 15 epochs for pre-training on COD10K with an initial learning rate of 1e − 5, and the learning rate decays by a factor of 10 after 10 epochs of training. Then, we load the pre-training weights on the COD10K dataset and perform 100 epochs for training. The initial learning rate is set to 1e − 4, and the learning rate decays by a factor of 10 after 50 epochs of training. The inter-frame interval is set to 4. The entire model

is trained in an end-to-end manner on an NVIDIA 3090 GPU. All input images are resized to 352 × 352 after random flipping, random rotation, and color dithering. The Adam optimizer is applied to optimize the model parameters.

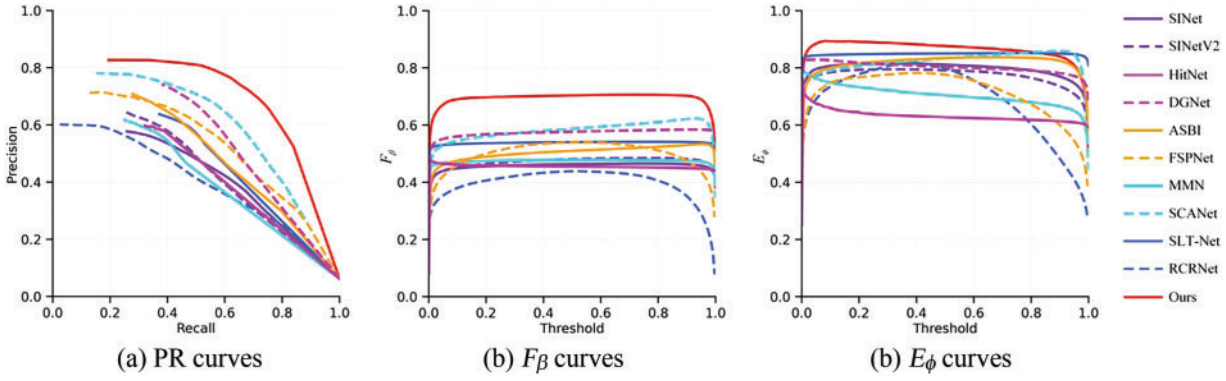
4.2 Comparisons with Other Methods

We conduct a comparative analysis of our proposed method with several recent image-based and video-based methods, including SINet [3], SINetV2 [4], HitNet [17], ASBI [23], DGNet [19], FSPNet [5], RCRNet [35], SCANet [36] MMN [11], STL-Net [7], and EMIP [8]. To ensure fair comparisons, the results of all these methods are generated by models retrained using their publicly available code. In cases where no publicly available code or results are difficult to obtain, we directly cite the results from the corresponding papers. The quantitative results for all methods are summarized in Table 1, and the qualitative performance is shown in Figs. 4 and 5.

Table 1: Quantitative comparisons with state-of-the-art methods on MoCA-Mask and CAD datasets

Method	Model	MoCA-Mask						CAD					
		$S_\alpha\uparrow$	$F_\beta^\omega\uparrow$	$M\downarrow$	$E_\phi\uparrow$	**mDic↑**	**mIoU↑**	$S_\alpha\uparrow$	$F_\beta^\omega\uparrow$	$M\downarrow$	$E_\phi\uparrow$	**mDic↑**	**mIoU↑**
Image-based	SINet [3]	0.605	0.231	0.021	0.669	0.276	0.202	0.665	0.403	0.042	0.794	0.439	0.334
	SINetV2 [4]	0.596	0.206	0.024	0.623	0.245	0.181	0.658	0.412	0.045	0.771	0.447	0.340
	HitNet [17]	0.572	0.179	0.010	0.488	0.182	0.150	0.639	0.397	0.036	0.631	0.402	0.314
	ASBI [23]	0.598	0.213	0.020	0.708	0.255	0.194	0.665	0.443	0.046	0.815	0.490	0.370
	DGNet [19]	0.583	0.198	0.027	0.666	0.232	0.170	0.705	0.511	0.038	0.800	0.537	0.414
	FSPNet [5]	0.588	0.151	0.044	0.593	0.201	0.147	0.699	0.434	0.046	0.713	0.468	0.363
Video-based	RCRNet [35]	0.558	0.138	0.026	0.536	0.165	0.113	0.656	0.345	0.052	0.693	0.372	0.277
	SCANet [36]	0.624	0.261	0.013	0.703	0.296	0.217	0.717	0.521	0.038	0.820	0.570	0.443
	MMN [11]	0.547	0.130	0.029	0.554	0.150	0.110	0.654	0.414	0.037	0.702	0.432	0.334
	SLT-Net [7]	0.631	0.311	0.027	0.759	0.360	0.272	0.696	0.481	0.030	0.845	0.493	0.402
	EMIP [8]	0.675	0.381	0.015	–	0.426	0.333	0.719	0.514	0.028	–	0.536	0.425
	Ours	0.682	0.401	0.012	0.777	0.422	0.345	0.767	0.646	0.025	0.862	0.540	0.534

Note: "↑"/"↓" indicates that larger/smaller is better. The best three results are highlighted in red, green and blue.



(a) PR curves (b) $F\beta$ curves (b) E_ϕ curves

Figure 4: PR, F_β and E_ϕ curves on the CAD dataset

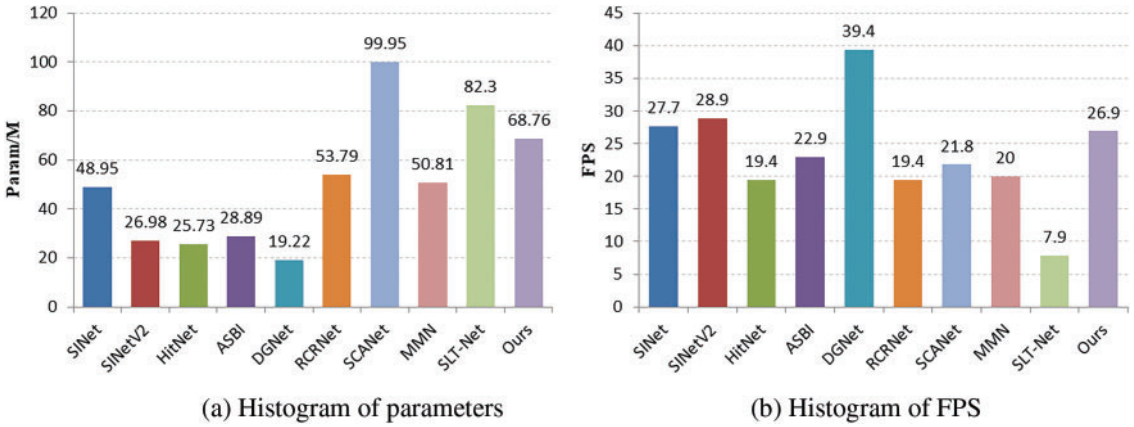


Figure 5: Number of parameters and FPS for different methods

1) Quantitative Comparison: Table 1 shows the detailed comparison results of each method on different datasets. On the MoCA-Mask test set, our method significantly outperforms recent methods, notably by 2% and 1.2% on metric F_β^ω and metric mIoU, respectively, compared to the best method EMIP in this evaluation. On the CAD dataset, our method obtains the best performance in five evaluation metrics except mDic, further validating its generalization ability. Besides, we show the PR, F_β and E_ϕ curves of the comparison methods on CAD dataset in Fig. 4 to provide a more comprehensive evaluation. Note that the higher the curve, the better the model performance. It is clear that our method (red curves) is better than other competitors. Fig. 5 provides a detailed comparison of the number of model parameters and the inference speed (FPS) of each method. As the method proposed in this paper requires a complex network structure to achieve effective extraction of spatiotemporal features, its number of parameters is relatively large, but still lower than the comparison methods such as SCANet and SLT-Net. Notably, despite the relative complexity of the model structure, our method demonstrates excellent performance in inference speed, with its frames per second (FPS) processing not only outperforming most of the comparative methods, but also reaching more than 25 FPS, which meets the basic requirement of real-time processing in practical applications. This result shows that the proposed method achieves a good balance between model efficiency and practicality.

2) Qualitative Comparison: Visual comparison of different methods on several typical samples are shown in Figs. 6 and 7. They present some challenging cases (sharp torsos, complex appearance textures, or blurred boundaries) in Fig. 6 and different states (moving or stationary) in Fig. 7. These results intuitively demonstrate the superior performance of the proposed method. Specifically, as shown in Fig. 6, our method can accurately localize and segment camouflaged objects with clear boundaries in complex contexts. As presented in Fig. 7, whether camouflaged objects are in motion or at rest, our method performs well in separating them from their surroundings. This success can be attributed to the effective utilization of temporal and spatial information.

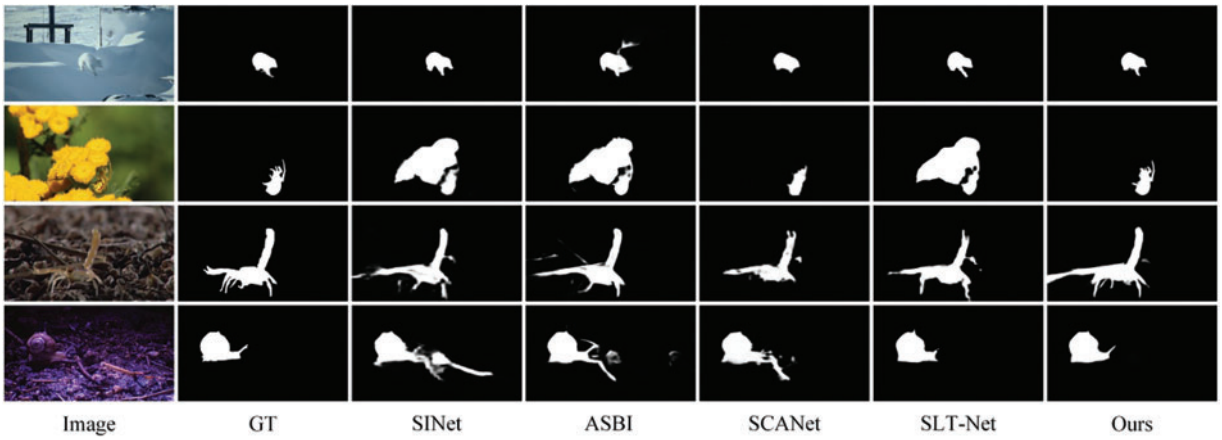


Figure 6: Visual comparisons of some recent methods and ours on different types of samples

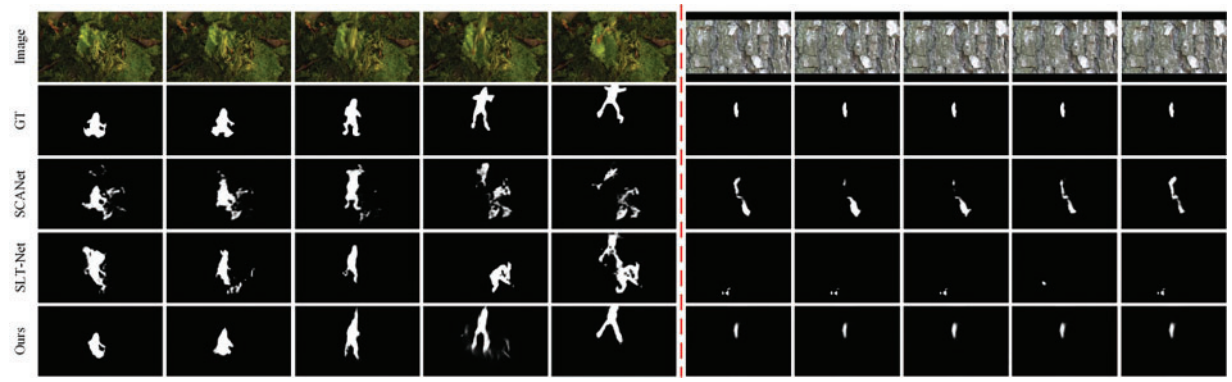


Figure 7: Visual comparison of some recent methods and ours on a sequence of frames spaced at intervals of 4

4.3 Ablation Studies

In this section, we perform a comprehensive ablation analysis to validate the effectiveness of our proposed modules.

In order to systematically evaluate the contribution of each key module to the network detection performance, we constructed a series of simplified models by removing SSAM, MFFM, and multi-task detection header modules one at a time, and performed a complete training and testing process on them. Table 2 presents the quantitative evaluation results for each simplified model in detail. The experimental data show that removing any of the modules resulted in varying degrees of model performance degradation, which strongly confirms the positive contribution of each module to improving detection performance. Particularly noteworthy is that the removal of the MFFM module results in the most significant decrease in model performance, a phenomenon that can be attributed to the multiple deployments of the MFFM at three different layers of the network, the cumulative effect of which makes the absence of this module have a more pronounced impact on the overall performance.

Table 2: Quantitative results of ablation experiments. M-Head indicates Multi-task Detection Head

SSAM	MFFM	M-Head	MoCA-Mask				CAD			
			$S_\alpha\uparrow$	$F_\beta^\omega\uparrow$	$M\downarrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$	$F_\beta^\omega\uparrow$	$M\downarrow$	$E_\phi\uparrow$
√	√	×	0.657	0.324	0.023	0.735	0.743	0.586	0.037	0.846
√	×	√	0.557	0.310	0.032	0.646	0.676	0.469	0.040	0.720
×	√	√	0.657	0.333	0.018	0.767	0.752	0.600	0.034	0.850
√	√	√	0.682	0.401	0.012	0.777	0.767	0.646	0.025	0.862

1) Effectiveness of SSAM: As described in Section 3.2, our SSAM integrates temporal features and spatial features with full consideration of their correlation and difference, enabling efficient fusion of different type features and obtaining more effective fusion features. To validate the effectiveness of SSAM, we conduct an ablation study using the following four variants obtained by changing the fusion method, and the simplified structures of the different variants are shown in Fig. 8.

i) "SSAM-A" uses simple element-wise addition to fuse temporal and spatial features.

ii) "SSAM-B" employs concatenation to fuse temporal and spatial features.

iii) "SSAM-C" applies element-wise addition to fuse enhanced temporal and spatial features.

(iv) "SSAM-D" uses concatenation to fuse enhanced temporal and spatial features.

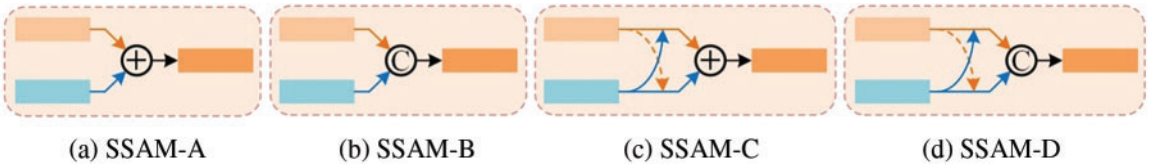


Figure 8: Structure of different SSAM variants

The quantitative results of the models equipped with SSAM or its variants are shown in Table 3. The model equipped with SSAM-C or SSAM-D outperforms the model equipped with SSAM-A or SSAM-B in all metrics, which validates the efficacy of enhanced learning between temporal and spatial features. Besides, the results in the bottom three rows of Table 3 indicate that biased fusion can better exploit the advantages of different features for VCOD than simple element-wise addition or concatenation operations. Overall, the model equipped with SSAM achieves the best performance among the various networks, which sufficiently demonstrates the effectiveness of our SSAM design. To more intuitively demonstrate the effect of SSAM, we visualize its input feature E_Q^{res5} and output feature $\widetilde{E}$. The results are shown in Fig. 9. By comparing the third and fourth columns of Fig. 9, we can clearly observe that SSAM effectively suppresses the background noise while significantly enhancing the positional salience of the camouflaged objects. This visualization result further corroborates the excellent performance of SSAM in fusing temporal and spatial features, providing intuitive and strong support for our method.

Table 3: The SSAM verification on MoCA-Mask dataset and CAD dataset

Method	MoCA-Mask				CAD			
	$S_\alpha\uparrow$	$F_\beta^\omega\uparrow$	$M\downarrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$	$F_\beta^\omega\uparrow$	$M\downarrow$	$E_\phi\uparrow$
SSAM-A	0.657	0.333	0.018	0.767	0.752	0.600	0.034	0.850
SSAM-B	0.657	0.329	0.018	0.763	0.752	0.599	0.035	0.850
SSAM-C	0.674	0.357	0.016	0.774	0.757	0.614	0.031	0.856
SSAM-D	0.673	0.354	0.017	0.771	0.755	0.610	0.032	0.853
Ours	0.682	0.401	0.012	0.777	0.767	0.646	0.025	0.862

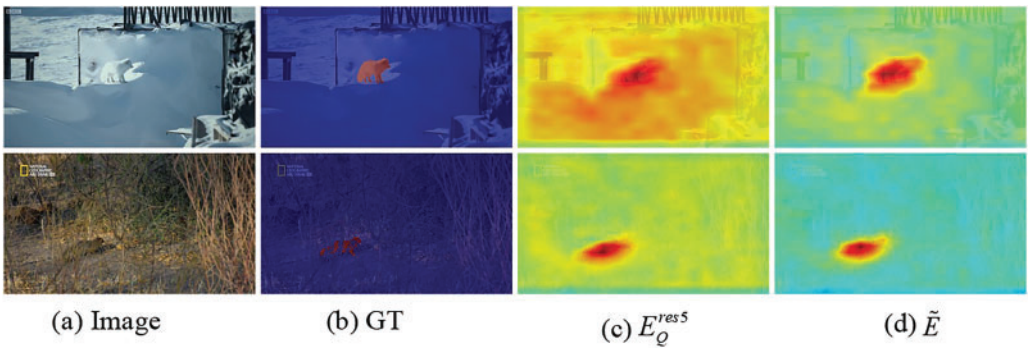


Figure 9: Feature visualization graphs in SSAM

2) Effectiveness of MFFM: As shown in Figs. 1 and 3, the Multi-Feature Fusion Module (MFFM) at the i-th layer ($i \in \{2,3,4\}$) refines its feature map (E_Q^{resi}) by considering additional three inputs: high-level features (f_s^{i+1} or $\widetilde{E}$), mask features (f_m^{i+1}), and boundary features (f_e^{i+1}). To evaluate the effectiveness of MFFM, we perform an ablation study by simplifying the proposed model in the following five cases:

i) "model-A": we remove all MFFMs from our model and make predictions directly on E_Q^{res2}, E_Q^{res3} and E_Q^{res4}.

ii) "model-B": we add MFFMs to layers 2, 3, and 4, but refine features only using high-level features (f_s^{i+1} or $\widetilde{E}$).

iii) "model-C": we keep all the MFFMs of our model, but the MFFM only employs high-level features (f_s^{i+1} or $\widetilde{E}$) and mask features (f_m^{i+1}) to refine the feature map.

iv) "model-D": we construct the model following model-C and add local channel attention to highlight important features.

v) "model-E": we add boundary features (f_e^{i+1}) to model-C to further refine the feature map, i.e., remove local channel attention from MFFM.

Table 4 reports the quantitative results for our full model and the above variants. Combining the additional three features (i.e., high-level features, mask features, and boundary features) together in MFFM yields better VCOD performance since, in Table 4, model-E achieves better metric results than the other three variants (i.e., model-A, model-B, and model-C). When comparing the MFFM with local channel attention (fourth and sixth rows) to the variants without local channel attention (third and fifth rows), the

model with local channel attention achieves better results, indicating that focusing on the important features in the MFFM helps our method identify the camouflaged objects more effectively. In addition, to further understand the effectiveness of our MFFM, some visualization results are shown in Fig. 10. By comparing the 3rd–5th and 7th columns, it can be noticed that the overall completeness of objects in the prediction results increases with the addition of additional features, and the clarity of objects' boundaries is significantly improved after the introduction of edge features. By comparing the 5th and 6th columns or the 7th and 8th columns, it is obvious that the fuzzy regions are dramatically reduced and the accuracy of objects' regions is further improved. In summary, the model with complete MFFM achieves the best performance in both quantitative and qualitative aspects, which proves the rationality and effectiveness of our MFFM design.

Table 4: The ablation verification of MFFM on MoCA-Mask dataset and CAD dataset

Method	MoCA-Mask				CAD			
	$S_\alpha\uparrow$	$F_\beta^\omega\uparrow$	$M\downarrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$	$F_\beta^\omega\uparrow$	$M\downarrow$	$E_\phi\uparrow$
model-A	0.557	0.310	0.032	0.646	0.676	0.469	0.040	0.720
model-B	0.665	0.370	0.022	0.751	0.756	0.634	0.029	0.852
model-C	0.671	0.380	0.014	0.772	0.761	0.639	0.025	0.850
model-D	0.676	0.399	0.014	0.774	0.764	0.640	0.027	0.857
model-E	0.675	0.399	0.014	0.772	0.761	0.636	0.025	0.855
Ours	0.682	0.401	0.012	0.777	0.767	0.646	0.025	0.862

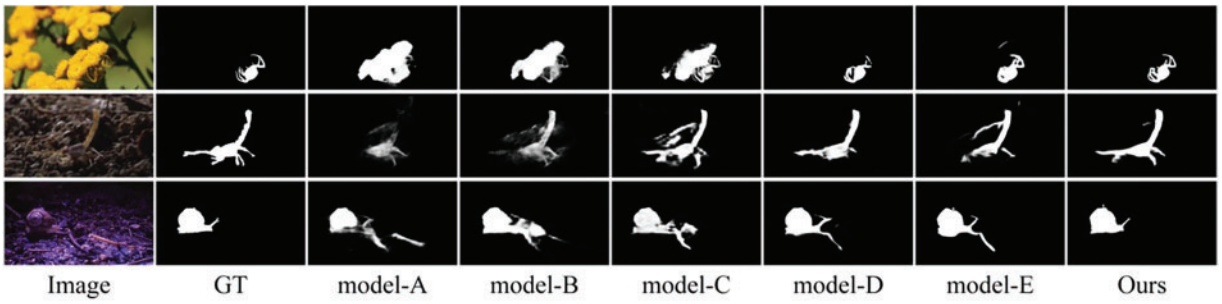


Figure 10: Some visual comparisons of the outputs of different variants

3) Effectiveness of Multi-task Detection Head: As mentioned in Section 3.4, our multi-task detection header establishes an explicit link between the segmentation branch and the boundary detection branch to enrich the feature representations of both branches. To verify the effectiveness of the multi-tasking detection head, we construct several detection head variants for ablation studies as follows, the structures of which are shown in Fig. 11.

i) "Head-A" uses only the segmentation branch.

ii) "Head-B" employs both branches without any interaction between them.

iii) "Head-C" allows information from the segmentation branch to pass to the boundary detection branch.

iv) "Head-D" allows information from the boundary detection branch to pass to the segmentation branch.

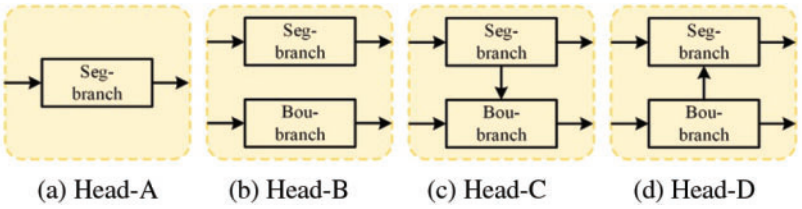


Figure 11: Structure of different multi-tasking detection head variants

Table 5 presents the quantitative detection results for models with different detection head variants. Even without inter-task interaction, Head-B outperforms the single-task model (Head-A), demonstrating the gains of multi-task learning. Models equipped with Head-C or Head-D performs better than Head-B, indicating that inter-task information transfer enhances performance and is beneficial for VCOD. Notably, the model with our proposed detection head achieves the best performance, showing that VCOD benefits from inter-task interaction. These results indicate that our model better leverages the mutually beneficial information between tasks compared to models with no interaction or only unidirectional information transfer.

Table 5: The multi-task detection head verification on MoCA-Mask dataset and CAD dataset

Method	MoCA-Mask				CAD			
	$S_\alpha\uparrow$	$F_\beta^\omega\uparrow$	$M\downarrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$	$F_\beta^\omega\uparrow$	$M\downarrow$	$E_\phi\uparrow$
Head-A	0.657	0.324	0.023	0.735	0.743	0.586	0.037	0.846
Head-B	0.668	0.348	0.017	0.766	0.756	0.608	0.033	0.853
Head-C	0.673	0.358	0.015	0.772	0.760	0.613	0.030	0.856
Head-D	0.677	0.361	0.014	0.775	0.763	0.619	0.028	0.859
Ours	0.682	0.401	0.012	0.777	0.767	0.646	0.025	0.862

Fig. 12 visualizes the impact of different detection heads on the model performance. By comparing the detection results between the second and third columns, it can be observed that the introduction of the boundary detection branch significantly enhances the model's target recognition capability, resulting in a significant improvement in the detection performance. Further analysis of the results in the third, fourth and fifth columns shows that the information transfer mechanism between different branches effectively improves the model's localization accuracy of the overall contour and boundary details of the target, where the information flow from the boundary branch to the segmentation branch brings about a more significant performance gain. Notably, by comparing the sixth column with other columns, it can be clearly found that the optimal detection effect is achieved by the detector head architecture with the bidirectional interaction mechanism between branches. This qualitative observation highly aligns with the previous quantitative evaluation results, further validating the effectiveness of the proposed detection head design.

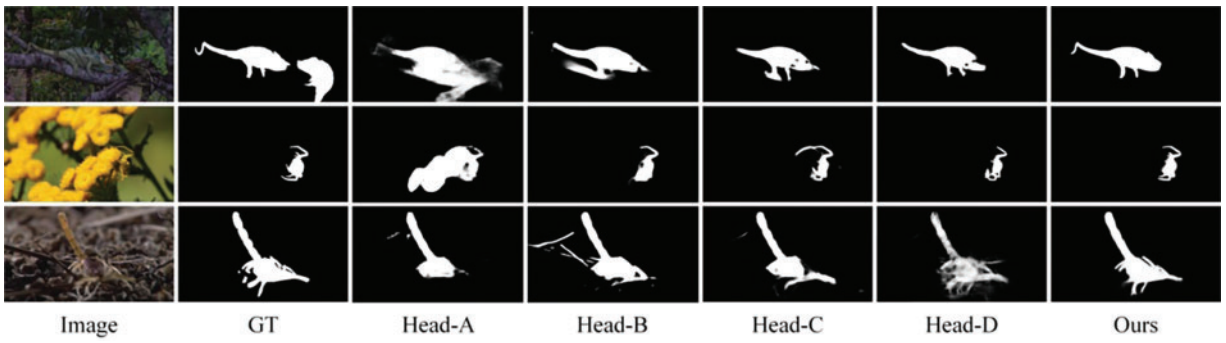


Figure 12: Detection results of models equipped with different detection heads

4.4 Limitations

We present some representative failure cases along with the results from the two SOTA methods in Fig. 13. Although the quantization performance of our method is satisfactory, it may fail in dim scenes or long-range camouflaged object detection. Specifically, as shown in the first row of Fig. 13, in a dim scene with low lighting, the model tends to confuse the object with the background due to fewer available cues in the space. As shown in the second row of Fig. 13, when the object is at a long distance, the object occupies a smaller area in the image. Motion cues are difficult to capture and the object is easily overwhelmed by the background, which leads to the model failing to accurately localize the camouflaged object. As can be seen from Fig. 13, such cases also easily confuse SOTA methods, so it is worth exploring further.

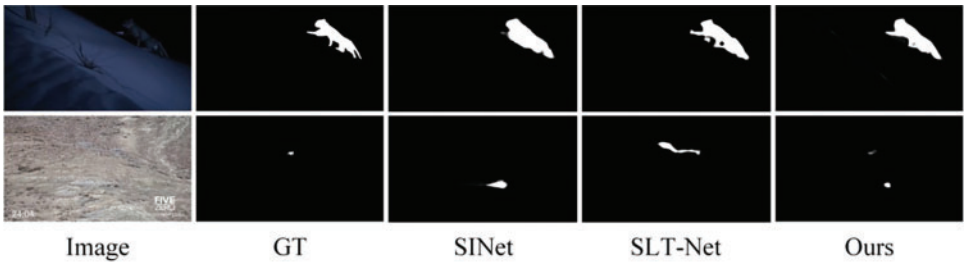


Figure 13: Representative failure cases

5 Conclusion

In contrast to previous methods, our approach in this paper emphasizes the efficient utilization of both temporal and spatial information for accurate detection of camouflaged targets in video sequences. Specifically, we first introduce STM into VCOD to obtain temporal information. Then, we propose the selective space-time aggregation module (SSAM), which enables collaborative learning and biased fusion of temporal and spatial information, resulting in spatiotemporal features with strong representational capabilities. Subsequently, drawing inspiration from the stepwise refinement strategy often adopted in ICOD, we develop the multi-feature fusion module (MFFM) to refine the camouflaged objects in a coarse-to-fine manner by exploiting the complementary nature of multi-type features. Finally, we introduce the multi-task detection header to prompt the model to benefit from multi-task learning. Our method achieves impressive results on the MoCA-Mask dataset and outperforms existing SOTA methods on the CAD dataset. In addition, we provide comprehensive ablation studies showing that the design of our modules is reasonable and effective.

Acknowledgement: We would like to express our sincere gratitude to Yan Xu and Changfeng Feng for their invaluable contributions to the dataset collection and production.

Funding Statement: Due to the confidentiality of the content of the projects, the names of the funded projects cannot be disclosed.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Dongdong Zhang, Chunping Wang; data collection: Qiang Fu, Huiying Wang; analysis and interpretation of results: Dongdong Zhang, Qiang Fu, Huiying Wang; draft manuscript preparation: Dongdong Zhang. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding author, Q. F., upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Zhao ZQ, Zheng P, Xu ST, Wu XD. Object detection with deep learning: a review. IEEE Transact Neural Netw Learn Syst. 2019;30(11):3212–32.
2. Zhuge MC, Fan DP, Liu N, Zhang DW, Xu D, Shao L. Salient object detection via integrity learning. IEEE Transact Pattern Anal Mach Intell. 2023;45(3):3738–52. doi:10.1109/TPAMI.2022.3179526.
3. Fan D, Ji G, Sun G, Cheng MM, Shen JB, Shao L. Camouflaged object detection. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020; Seattle, WA, USA. p. 2774–84.
4. Fan DP, Ji GP, Cheng MM, Shao L. Concealed object detection. IEEE Transact Pattern Anal Mach Intell. 2022;44(10):6024–42. doi:10.1109/TPAMI.2021.308576.
5. Huang Z, Dai H, Xiang TZ, Wang S, Chen HX, Qin J, et al. Feature shrinkage pyramid for camouflaged object detection with transformers. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2023; Vancouver, BC, Canada: IEEE. p. 5557–66.
6. Lamdouar H, Yang C, Xie W, Zisserman A. Betrayed by motion: camouflaged object discovery via motion segmentation. In: Proceedings of the Asian Conference on Computer Vision (ACCV); 2020; Kyoto, Japan: Springer.
7. Cheng XL, Xiong H, Fan DP, Zhong YR, Harandi M, Drummond T, et al. Implicit motion handling for video camouflaged object detection. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022; New Orleans, LA, USA: IEEE. p. 13854–63.
8. Zhang X, Xiao T, Ji GP, Xuan W, Fu K, Zhao QJ. Explicit motion handling and interactive prompting for video camouflaged object detection. arXiv:2403.01968. 2024.
9. Chen CLZ, Wang GT, Peng C, Fang YM, Zhang DW, Qin H. Exploring rich and efficient spatial temporal interactions for real-time video salient object detection. IEEE Trans Image Process. 2021;30:3995–4007. doi:10.1109/TIP.2021.3068644.
10. Oh SW, Lee JY, Xu N, Kim SJ. Video object segmentation using space-time memory networks. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV); 2019; Seoul, Republic of Korea. p. 9225–34.
11. Zhao X, Liang HR, Li PP, Sun GD, Zhao DD, Liang RH, et al. Motion-aware memory network for fast video salient object detection. IEEE Transact Image Process. 2024;33:709–21. doi:10.1109/TIP.2023.3348659.
12. Boot WR, Neider MB, Kramer AF. Training and transfer of training in the search for camouflaged targets. Attent Percept Psycho. 2009;71:950–63. doi:10.3758/APP.71.4.950.
13. Galun, Sharon, Basri, Brandt. Texture segmentation by multiscale aggregation of filter responses and shape elements. In: Proceedings Ninth IEEE International Conference on Computer Vision; 2003; Nice, France: IEEE. p. 716–23.

14. Lu HM, Wang XC, Liu SZ, Liu SZ, Shi MD, Guo AK. The possible mechanism underlying visual anti-camouflage: a model and its real-time simulation. IEEE Transact Syst Man Cybernet-Part A: Syst Hum. 1999;29(3):314–8. doi:10.1109/3468.759290.
15. Le T, Nguyen TV, Nie ZL, Tran MT, Sugimoto A. Anabranch network for camouflaged object segmentation. Comput Vis Image Underst. 2019;184:45–56. doi:10.1016/j.cviu.2019.04.006.
16. Lv YQ, Zhang J, Dai YC, Li AX, Liu BW, Barnes N, et al. Simultaneously localize, segment and rank the camouflaged objects. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021 June 19–25. Piscataway, NJ, USA: IEEE Computer Society. p. 11591–601.
17. Hu XB, Wang S, Qin XB, Dai H, Ren WQ, Luo DH, et al. High-resolution iterative feedback network for camouflaged object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence; 2023; Washington, DC, USA: AAAI. p. 881–9.
18. Zhuge MC, Lu XK, Guo YY, Cai ZH, Chen SH. Cubenet: X-shape connection for camouflaged object detection. Pattern Recognit. 2022;127:108644. doi:10.1016/j.patcog.2022.108644.
19. Ji GP, Fan DP, Chou YC, Dai DX, Liniger A, Van GL. Deep gradient learning for efficient camouflaged object detection. Mach Intell Res. 2023;20(1):92–108. doi:10.1007/s11633-022-1365-9.
20. Zhong YJ, Li B, Tang L, Kuang SY, Wu S, Ding SH. Detecting camouflaged object in frequency domain. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022; New Orleans, LA, USA: IEEE. p. 4504–13.
21. Qin XB, Fan DP, Huang CY, Diagne C, Sant'Anna AAC, Suarez A, et al. Boundary-aware segmentation network for mobile and web applications. arXiv:2101.04704. 2021.
22. Ji GP, Zhu L, Zhuge MC, Fu KR. Fast camouflaged object detection via edge-based reversible re-calibration network. Pattern Recognit. 2022;123(6):108414. doi:10.1016/j.patcog.2021.108414.
23. Zhang Q, Sun XX, Chen YR, Ge YL, Bi HB. Attention-induced semantic and boundary interaction network for camouflaged object detection. Comput Vis Image Underst. 2023;233(8):103719. doi:10.1016/j.cviu.2023.103719.
24. Zhai Q, Li X, Yang F, Chen C.L. Z, Cheng H, Fan DP. Mutual graph learning for camouflaged object detection. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2021; Nashville, TN, USA: IEEE. p. 12997–13007.
25. Zhang HL, Liang JM, Zhang JP, Zhang TZ, Lin YZ, Wang YF. Attention-driven memory network for online visual tracking. IEEE Transact Neural Netw Learn Syst. 2024;35(12):17085–98. doi:10.1109/TNNLS.2023.3299412.
26. Li YL, Zhang TZ, Liu X, Tian Q, Zhang YD, Wu F. Visible-infrared person re-identification with modality-specific memory network. IEEE Trans Image Process. 2022;31:7165–78. doi:10.1109/TIP.2022.3220408.
27. Zhang XD, Xiao Y, Peng JY, Gao XB, Hu B. Confidence-based dynamic cross-modal memory network for image aesthetic assessment. Pattern Recognit. 2024;149(2):110227. doi:10.1016/j.patcog.2023.110227.
28. Cheng HK, Tai YW, Tang CK. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. Adv Neural Inf Process Syst. 2021;34:11781–94.
29. Liang SX, Shen X, Huang JQ, Hua XS. Video object segmentation with dynamic memory networks and adaptive object alignment. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV); 2021; Montreal, QC, Canada: IEEE. p. 8045–54.
30. Shaker A, Wasim ST, Danelljan M, Khan S, Yang MH, Khan FS. Efficient video object segmentation via modulated cross-attention memory. arXiv:2403.17937. 2024.
31. He KM, Zhang XY, Ren SQ, Sun J. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016; Las Vegas, NV, USA: IEEE. p. 770–8.
32. Cheng TH, Wang XG. Boundary-preserving mask R-CNN. In: Computer Vision–ECCV 2020: 16th European Conference; 2020; Glasgow, UK. p. 660–76.
33. Bideau PALE. It's Moving! a probabilistic model for causal motion segmentation in moving camera videos. In: Computer Vision–ECCV 2016 14th European Conference; 2016; Amsterdam, The Netherlands. p. 433–49.
34. Perazzi F, Krähenbühl P, Pritch Y, Hornung A. Saliency filters: contrast based filtering for salient region detection. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition; 2012; Providence, RI, USA: IEEE; p. 733–40.

35. Yan PX, Li GB, Xie Y, Li Z, Wang C, Chen TS, et al. Semi-supervised video salient object detection using pseudo-labels. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV); 2019; Seoul, Republic of Korea. p. 7283–92.
36. Chen TY, Xiao J, Hu XG, Zhang GF, Wang SJ. Spatiotemporal context-aware network for video salient object detection. Neural Comput Applicat. 2022;34(19):16861–77. doi:10.1007/s00521-022-07330-1.