Adaptive Kernel Selection Module Combined With Feature Enhanced Perception Network for Camouflaged Object Detection

Ruyu Liu, *Member, IEEE*, Guang Yang, Feng Xiao, *Student Member, IEEE*, Jianhua Zhang, *Senior Member, IEEE*, and Shengyong Chen, *Senior Member, IEEE*

***Abstract*—Camouflaged object detection plays a crucial role in applications such as automatic sorting and defect inspection in industrial production, yet existing methods often struggle to flexibly capture features of diverse shapes, orientations, and scales due to their reliance on fixed receptive fields and rigid windowing schemes. To address these limitations, we propose a dual-branch joint network comprising a reference branch and a segmentation branch. The reference branch learns supplementary cues from salient objects that co-occur with camouflaged targets, guiding the segmentation branch toward more accurate delineation. Within the segmentation branch, we introduce three novel modules: 1) a deformable window interaction mechanism that replaces fixed-size transformer windows with learnable quadrilateral windows to adaptively extract features of arbitrary shape and orientation; 2) a feature enhancement perception module that fuses rich multiscale representations through parallel dilated convolutions at varying rates and channel-/spatial-attention mechanisms; and 3) a receptive field adjustment adaptive module that dynamically adjusts its receptive field size to balance sensitivity to fine details and global context. Comprehensive experiments on COD10 K, NC4K, CAMO, and R2C7K benchmarks demonstrate that our model outperforms the majority of current state-of-the-art approaches, while ablation studies and sensitivity analyses confirm the individual and combined effectiveness of our proposed components.**

***Index Terms*—Camouflaged object detection, computer vision, deep learning, salient object detection (SOD), vision transformer.**

I. Introduction

CAMOUFLAGE object detection aims to segment objects concealed in the environment that are not easily discernible. Camouflaged target detection can be applied in the fields of quality inspection, safety monitoring, material classification, and recycling in industrial production. In the early stages, detection of camouflaged objects relied heavily on manually crafted features such as brightness, color, and texture, but these methods lacked broad applicability [1]. Later, with the development of deep learning and the emergence of large-scale datasets [2], many deep learning-based camouflaged object detection methods were proposed. With the availability of large amounts of data for training, these methods are able to automatically extract features and detect camouflaged objects, achieving excellent performance. Recently, some researchers have successfully applied techniques such as gradient-based methods [3], boundary-guided methods [4], uncertainty-aware approaches [5], multiscale strategies [6], and multiview inputs [7]. Additionally, other methods that mimic human or predator behavior, such as amplification strategies [7] and two-stage strategies [8], have also achieved promising results.

Although existing methods have achieved satisfactory performance in camouflaged object detection [9], [10], [11], there are still two main challenges. First, existing transformer-based models typically use fixed-size window partitioning, but camouflaged objects in the environment have varying orientations and shapes. Fixed-size windows cannot flexibly capture features of different directions and shapes. Second, there is a significant size variation among camouflaged targets, and the required contextual information differs for objects of different sizes. Smaller targets require larger contextual information to be separated from the environment, while larger targets do not need such extensive context as the object itself contains enough information. In fact, overly large contextual information can mislead the model's inference. To address these challenges, we propose a dual-branch network consisting of a reference branch and a segmentation branch, using salient images corresponding to camouflaged objects as auxiliary information to assist the model in detecting camouflaged targets. Additionally, we introduce the deformable window Interaction mechanism (DWIM), feature enhancement perception (FEP) module, and receptive field adjustment adaptive (RFAA) module to tackle

Received 27 June 2025; accepted 7 August 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62202137, in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LMS25F020009. Paper no. TII-25-4307. *(Corresponding author: Ruyu Liu.)*

Ruyu Liu is with the School of Information Science and Technology, Hangzhou Normal University, Hangzhou 311121, China (e-mail: lry@hznu.edu.cn).

Feng Xiao is with the Intelligent Systems Laboratory, Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology, 6009 Ålesund, Norway (e-mail: fengx@ieee.org).

Guang Yang, Jianhua Zhang, and Shengyong Chen are with the Tianjin University of Technology, Tianjin, P.R. 300384, China (e-mail: yangguang@stud.tjut.edu.cn; zjh@ieee.org; sy@ieee.org).

Digital Object Identifier 10.1109/TII.2025.3609076

the aforementioned challenges. To validate the effectiveness of the proposed model, we conducted comparative experiments on the COD10 K, CAMO, NC4K, and R2C7K datasets, showing that our method outperforms the current state-of-the-art methods in the field of camouflaged object detection. We split the COD10 K and R2C7K datasets into different categories and varying amounts of data for further comparative experiments, and the results again demonstrate that our method outperforms others. In addition, we performed extensive ablation studies and sensitivity analyses, which further confirm the effectiveness and robustness of our method.

Overall, the contributions of this article can be summarized as follows.

1) We propose the DWIM, which transforms the fixed-size windows used in traditional vision transformers into arbitrary quadrilateral windows, enhancing the flexibility in extracting features of varying shapes and orientations.
2) We propose the FEP module, which captures rich multiscale features through dilated convolutions with different dilation rates and various attention mechanisms.
3) We introduce a RFAA module that dynamically adjusts the receptive field, expanding context for small targets to separate them from complex backgrounds and limiting it for larger ones to reduce noise, thus boosting detection accuracy and robustness across scales.

II. Related Work

In this section, we introduced the concepts and research development of salient object detection (SOD) and camouflaged object detection, respectively.

A. Salient Object Detection

SOD aims to identify and locate the most visually attractive objects or regions in an image. Early methods of COD largely drew inspiration from SOD. In the early stages, traditional methods for SOD largely relied on manually crafted features and heuristic rules [12]. Although these methods based on manually crafted features perform well in certain scenarios, the design and extraction of these features require a significant amount of expertise and experience. Moreover, they often need optimization for specific tasks. The advent of deep learning has propelled the evolution of SOD into a new phase [13], utilizes a fully convolutional neural network to detect salient regions in an image [14], adopt a multistage supervised approach [15], locates salient objects by combining depth information with multiscale contextual features [16], achieves excellent performance in predicting salient regions by introducing adversarial training into deep convolutional neural networks.

We utilize the results of SOD as reference information, enabling more targeted identification of specific camouflaged targets. Introducing SOD information in camouflaged object detection tasks provides clearer semantic guidance, thereby improving the accuracy of target localization and segmentation.

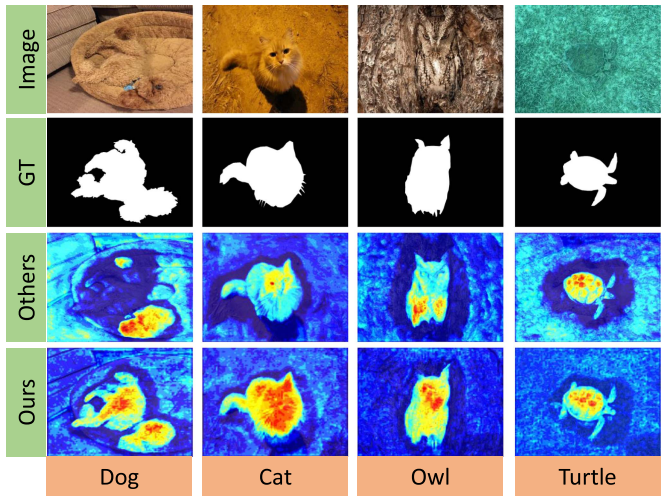

Fig. 1. Visualization results of class activation heatmaps for camouflaged objects of different sizes in our model.

B. Camouflaged Object Detection

Camouflaged object detection refers to the use of computer vision techniques to identify and locate objects that are difficult to distinguish in their surrounding environment. Camouflaged objects are hidden within the environment and are not easily detected. In the early stages, research primarily relied on handcrafted features, such as texture and color [17]. However, due to the limitations of handcrafted features, these methods were typically only applicable in relatively simple scenarios. In the era of data-driven deep learning, various methods have emerged, including multiscale strategies [10], [18], multiview approaches [7], uncertainty-based methods [5], [9], and amplification strategies [7] that mimic human behavior, as well as two-stage strategies [8]. There are also methods that use additional information to assist the model in segmentation, such as boundaries [4], salient images [19], etc. Inspired by these approaches, we also use the corresponding salient images of camouflaged targets as additional information to help the model learn the data distribution.

III. Method

In this section, we first introduce our network architecture in Section III-A. Subsequently, Sections III-B, III-C, and III-D provide detailed explanations of DWIM, the FEP module, and the RFAA module, respectively.

A. Network Architecture

Fig. 2 illustrates that our model consists of two branches: the reference branch and the segmentation branch. In the reference branch, taking K reference images with sizes $C \times H \times W$ as an example, where H is the height, W is the width, and C is the number of channels. Visual features of size $\frac{H}{32} \times \frac{W}{32}$ and foreground maps of size $H \times W$ are extracted from the encoder and decoder, respectively, we denote them as $\{F_k^{\text{ref}}\}_{k=1}^{K}$ and $\{M_k^{\text{ref}}\}_{k=1}^{K}$. Then they are fed into the masked average pooling

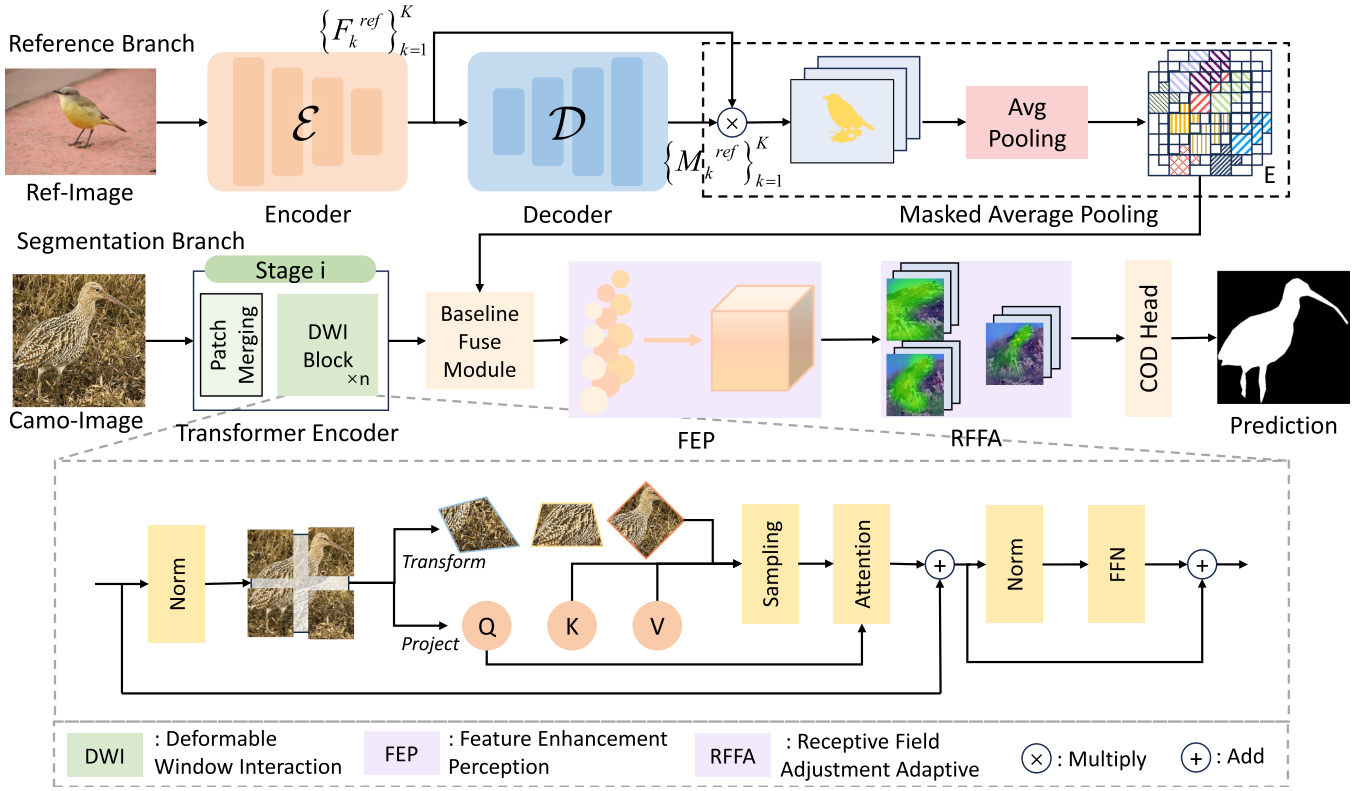

Fig. 2. Structure of our network includes a reference branch and a segmentation branch. The network takes camouflaged target images and their corresponding salient target images as input, and produces the final segmentation map as output.

module to obtain the shared representation E of the target object. The feature E extracted by the reference branch helps the segmentation branch focus on the key regions of the image while ignoring the background or irrelevant regions during camouflaged object segmentation. Therefore, the feature extracted by the reference branch serves as a "guidance" mechanism, enabling the model to better localize camouflaged objects and achieve accurate segmentation. This process can be described as follows:

$$E = \mathcal{F}_{\text{conv}}^{1\times1}\left(\frac{\mathcal{F}_{2d}\left(\mathcal{F}_{\text{down}}\left(M_k^{\text{ref}}\right)\otimes F_k^{\text{ref}}\right)}{\mathcal{F}_{2d}\left(F_k^{\text{ref}}\right)}\right) \tag{1}$$

where $\otimes$ is the elementwise multiplication operation, $\mathcal{F}_{\text{down}}(\cdot)$ is a bilinear downsampling operation for shape matching, $\mathcal{F}_{2d}(\cdot)$ accumulates the feature values along the spatial dimension, $\mathcal{F}_{\text{conv}}^{1\times1}(\cdot)$ transforms the channels into C_d, where in our experiments $C_d = 2048$, striking a balance between efficiency and performance. In the segmentation branch, we introduce the innovative deformable window interaction mechanism, which is incorporated into the transformer encoder, enabling the model to flexibly extract features from multiple directions and shapes. We also propose the FEP module and the RFAA module for adaptively extracting multiscale contextual information.

B. Deformable Window Interaction Mechanism

Since the transformer was adapted from the field of natural language processing to the vision domain, its efficiency in using the window-based attention mechanism has made it a widely used backbone network in visual tasks. Whether it is the traditional ViT, which divides the image into fixed-size patches, flattens each patch, and maps it into an embedding vector for global self-attention computation, or the swin transformer, which computes attention within local windows and then achieves global modeling through sliding windows, the window sizes they use are fixed and data-agnostic. We believe that such fixed window shapes limit the ability to extract features of objects with different sizes, orientations, and shapes in the input image.

To address the challenges posed by using fixed-size windows to handle various objects, we propose a DWIM and incorporate it into the transformer. Similar to the traditional window-based attention mechanism, we first divide the input feature map into predefined patches of size p, referred to as the base windows P, and then derive the query, key, and value from the window features

$$Q, K, V = Linear(P). \tag{2}$$

Next, we apply a projection transformation to process the base windows. The resulting quadrilaterals are highly flexible in terms of position, size, orientation, and shape, allowing them to cover objects of varying sizes, directions, and shapes. The projection matrix we use can be defined in the following

form:

$$T = \begin{pmatrix} a_1 & a_2 & b_1 \\ a_3 & a_4 & b_2 \\ c_1 & c_2 & 1 \end{pmatrix}. \tag{3}$$

The elements a_1, a_2, a_3, a_4 in the matrix define the scaling, rotation, and shearing of the points, controlling the magnification, rotation, and deformation of objects on the image plane. b_1, b_2 define the translation transformation, representing the amount of translation of the 3-D points along the image plane. The elements c_1 and c_2 in the matrix define the effect of the projection transformation, especially influencing the perspective depth. Directly obtaining these nine parameters is challenging, so we decompose the projection transformation into a series of basic transformations, such as scaling T_s, shearing T_h, rotation T_r, translation T_t, and projection T_p. To obtain these parameters, we propose a projection transformation parameter learning module. This module consists of an average pooling layer, an ReLU activation function, and a 1×1 convolution layer

$$T_s = \begin{pmatrix} t_1 + 1 & 0 & 0 \\ 0 & t_2 + 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, T_h = \begin{pmatrix} 1 & t_3 & 0 \\ t_4 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{4}$$

$$T_r = \begin{pmatrix} \cos t_5 & -\sin t_5 & 0 \\ \sin t_5 & \cos t_5 & 0 \\ 0 & 0 & 1 \end{pmatrix}, T_t = \begin{pmatrix} 1 & 0 & t_6 \\ 0 & 1 & t_7 \\ 0 & 0 & 1 \end{pmatrix} \tag{5}$$

$$T_p = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_8 & t_9 & 1 \end{pmatrix}, T = T_s \times T_h \times T_r \times T_t \times T_p. \tag{6}$$

We perform parallel computation on each point of the base window to obtain its final position after the projection transformation. This process can be represented as

$$\begin{aligned} \left[x_i^{tmp}, y_i^{tmp}, z_i^{tmp}\right]^T &= T \times [x_i, y_i, 1]^T \\ (x_i', y_i') &= \left(x_i^{tmp}/z_i^{tmp}, y_i^{tmp}/z_i^{tmp}\right). \end{aligned} \tag{7}$$

Here, x_i and y_i are the points in the base window, while x_i' and y_i' are the points after the window transformation.

However, we notice an issue. When generating quadrilaterals for each window, simply using absolute coordinates in the same coordinate system can lead to ambiguity. For example, given two windows at different positions, even if they share the same projection transformation matrix, the projection transformation effects corresponding to windows at different distances from the origin will differ. This will complicate the optimization of the projection transformation parameter learning module during training. To address this issue, we use the coordinates relative to the center of each window as the input for the projection transformation and transform them into the final coordinates after the computation is completed

$$\begin{aligned} (x_i^r, y_i^r) &= (x_i - x^c, y_i - y^c) \\ (x_i', y_i') &= (x_i^r + x^c, y_i^r + y^c). \end{aligned} \tag{8}$$

Here, x^c and y^c represent the coordinates of the center of the window, x_i^r and y_i^r represent the relative coordinates of a point within the window with respect to the center, and x_i' and y_i' represent the coordinates of the point after the window transformation.

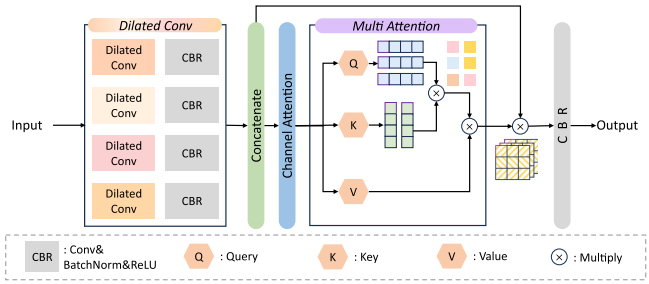

Fig. 3. Structural details of our FEP module.

C. Feature Enhancement Perception Module

Camouflaged targets are concealed within the background, with very subtle differences between them. Therefore, leveraging global information is crucial for improving the effectiveness of camouflaged object detection. To capture multiscale contextual information, we propose an FEP module and incorporate it into the segmentation branch.

As shown in Fig. 3, taking into account the output from the preceding module, which is represented as x. x undergoes parallel operations through four dilated convolutions with different sampling rates, followed by convolution layer, batch normalization and ReLU activation function. In this way, it can capture multiscale information from local to global

$$a_i = \mathcal{F}_{dc}^i(x), b_i = \mathcal{F}_{\text{relu}}(\mathcal{F}_{bn}(\mathcal{F}_{\text{conv}}(a_i))) \tag{9}$$

where $\mathcal{F}_{dc}(\cdot)$ represents dilated convolution, $\mathcal{F}_{\text{conv}}(\cdot)$ represents convolution, $\mathcal{F}_{bn}(\cdot)$ represents batch normalization, and $\mathcal{F}_{\text{relu}}(\cdot)$ represents the ReLU activation function. Subsequently, they are fused together, and this process can be formulated as

$$c = \mathcal{F}_{\text{con}}(b_i) \tag{10}$$

where $\mathcal{F}_{con}(\cdot)$ represents the concatenation operation. The concatenated features are then processed through channel attention and multihead attention, and the resulting output is multiplied elementwise with c. Afterward, it undergoes convolution, batch normalization, and ReLU activation to obtain the final output.

D. Receptive Field Adjustment Adaptive Module

Vision transformers have gained significant attention in image segmentation due to their large receptive fields, which are key to their success. Recent studies show that well-designed convolutional networks with large receptive fields can also compete with transformer models. Inspired by this, we propose a novel approach for camouflaged object detection. Our RFAA module dynamically adjusts the receptive field of features to better handle the varying contexts of camouflaged targets. This is done through a spatial selection mechanism that weights and merges features processed by a sequence of large depthwise kernels, with kernel weights dynamically determined based on the input.

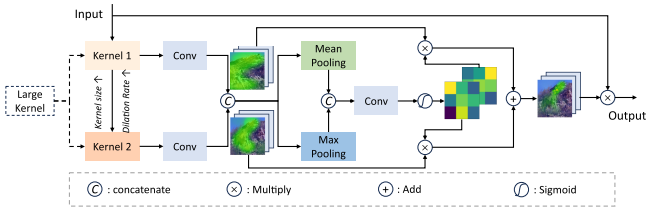

Fig. 4. Structural details of our RFAA module.

We first decompose large convolution kernels from other studies into multiple depthwise separable convolutions with gradually increasing kernel sizes and dilation rates. This approach explicitly generates multiple features with different receptive fields, making subsequent kernel selection easier. Furthermore, under the same theoretical receptive field, our method significantly reduces the number of parameters compared to traditional large convolution kernels. As shown in Fig. 4, we define the input as x, and the features obtained by processing with different kernels are denoted as a_i

$$a_0 = x, a_{i+1} = \mathcal{F}_p^i(\mathcal{F}_d^i(a_i)) \tag{11}$$

where $\mathcal{F}_d^i(\cdot)$ represents depthwise convolution, $\mathcal{F}_p^i(\cdot)$ represents pointwise convolution, and i denotes the index of the corresponding depthwise separable convolution. The obtained features a_i are further processed by a 1×1 convolution layer

$$b_i = \mathcal{F}_{\text{conv}}^{1\times1}(a_i). \tag{12}$$

Next, we concatenate the obtained features with different receptive fields and apply average pooling and maximum pooling to them separately

$$c = \mathcal{F}_{\text{mean}}(\mathcal{F}_{\text{con}}(b_i)), d = \mathcal{F}_{\text{max}}(\mathcal{F}_{\text{con}}(b_i)). \tag{13}$$

The features processed with different pooling methods are then concatenated again and passed through a convolution layer to change the number of channels, followed by a sigmoid activation to obtain the spatial attention maps

$$e = \mathcal{F}_{\text{sig}}(\mathcal{F}_{\text{con}}(c, d)). \tag{14}$$

Then, the features from the series of decomposed large convolution kernels are weighted by the corresponding spatial attention maps and fused through a convolution layer to obtain the attention features

$$f = \mathcal{F}_{\text{conv}}(\mathcal{F}_{\text{add}}(a_i, e_i)). \tag{15}$$

Finally, the attention features are elementwise multiplied with the input to produce the output

$$g = x \cdot f. \tag{16}$$

IV. Experiment

In this section, we first introduce the dataset used in the experiment. Subsequently, we discuss the experimental setup and results. Finally, we conduct ablation studies and sensitivity analyses to reveal the effectiveness of individual components in the model and examine the model's sensitivity and robustness to different factors in the input data.

A. Dataset Introduction

1) COD10 K Dataset: The dataset [2] contains 10 000 images, divided into 10 super-classes, and 78 subclasses which are collected from multiple photography websites.

2) R2C7K Dataset: The dataset [19] consists of a total of 6615 images, including both the Camo-subset and the Ref-subset. It covers 64 categories, such as dogs, turtles, and more.

B. Experimental Setup

1) Evaluation Metrics: We adopt mean absolute error (M), structure-measure (S_m), adaptive E-measure (αE), and weighted F-measure (wF) as metrics to evaluate the effectiveness of the model.

2) Loss Function: The loss function is formulated as a blend of binary cross-entropy loss (BCE) and intersection over union loss (IoU)

$$\text{Loss} = \sum_{i=1}^{4} \alpha Loss_{\text{bce}}(P_i, G) + \beta Loss_{\text{iou}}(P_i, G). \tag{17}$$

Here, P is generated by different modules of the model, and G represents the Ground Truth.

3) Hyperparameter Settings: In the training phase, we set the batch size to 32 and the number of reference images to 5. We use the Adam optimizer. The learning rate is initialized to 5e-4, and cosine annealing is applied with a decay period of 100 epochs. Images are resized to 352×352 before being input to the model. In the inference phase, all images are resized to dimensions of 352×352 and inputted into the model, generating the final predictions. All experiments in the later sections are conducted on an Nvidia RTX 3090 GPU and in the PyTorch 1.8.1 environment.

C. Experimental Results

To comprehensively validate the effectiveness of the model, we conducted extensive experiments along with quantitative and qualitative analyses.

1) Comparison With Other Camouflaged Object Detection Methods on COD10 K Dataset and R2C7K Dataset: To assess the efficiency of our model, we selected several recent, representative, and open-source models for performance comparison on the COD10 K dataset and R2C7K dataset. The chosen models should satisfy the following conditions: a) recently published, b) representative, and c) open-source. Ultimately, the chosen models include: PFNet [13], SINetV2 [6], PreyNet [20], BSANet [21], BGNet [22], ZoomNet [23], DGNet [24], CamFormer [9], and R2CNet [19]. The evaluation metrics (M, S_m, αE, wF) obtained in experiments are presented in Table I, indicating that our model outperforms the others. Additionally, the segmentation results of our model for camouflaged objects, as well as the visualization of the label values, are shown in Fig. 5.

2) Comparison With Other Camouflaged Object Detection Methods on CAMO Dataset and NC4K Dataset: To evaluate the effectiveness of our approach, we compare it against a suite of

TABLE I
QUANTITATIVE COMPARISON WITH OTHER METHODS THAT INCORPORATE REFERENCE BRANCHES (DENOTED AS †), WHERE "↑" INDICATE THAT HIGHER VALUES ARE BETTER, WHEREAS "↓" INDICATE THAT LOWER VALUES ARE PREFERABLE; THE TOP TWO RESULTS OF EACH METRIC ARE HIGHLIGHTED IN **RED** AND **BLUE**.

Method	Origin	COD10K				R2C7K			
		$M \downarrow$	$S_m \uparrow$	$\alpha E \uparrow$	$wF \uparrow$	$M \downarrow$	$S_m \uparrow$	$\alpha E \uparrow$	$wF \uparrow$
PFNet † [13]	CVPR'21	0.036	0.802	0.881	0.623	0.038	0.794	0.877	0.609
SINetV2 † [20]	TPAMI'22	0.035	0.817	0.889	0.686	0.036	0.813	0.874	0.678
PreyNet † [20]	MM'22	0.033	0.813	0.891	0.699	0.034	0.809	0.890	0.693
BSANet † [21]	AAAI'22	0.034	0.823	0.896	0.715	0.034	0.819	0.893	**0.703**
BGNet † [22]	IJCAI'22	0.033	0.826	**0.907**	0.699	0.036	**0.820**	**0.903**	0.681
ZoomNet † [23]	CVPR'22	**0.030**	0.821	0.891	0.721	**0.032**	0.817	0.886	0.690
DGNet † [24]	MIR'23	0.032	**0.827**	0.894	**0.723**	**0.033**	**0.820**	0.885	0.689
CamoFormer † [9]	TPAMI'24	0.029	0.838	0.898	0.730	-	-	-	-
R2CNet † [19]	TPAMI'25	0.033	0.819	0.899	0.712	0.035	0.809	0.883	0.677
Ours	TII'2025	**0.029**	**0.847**	**0.912**	**0.731**	**0.032**	**0.831**	**0.894**	**0.707**

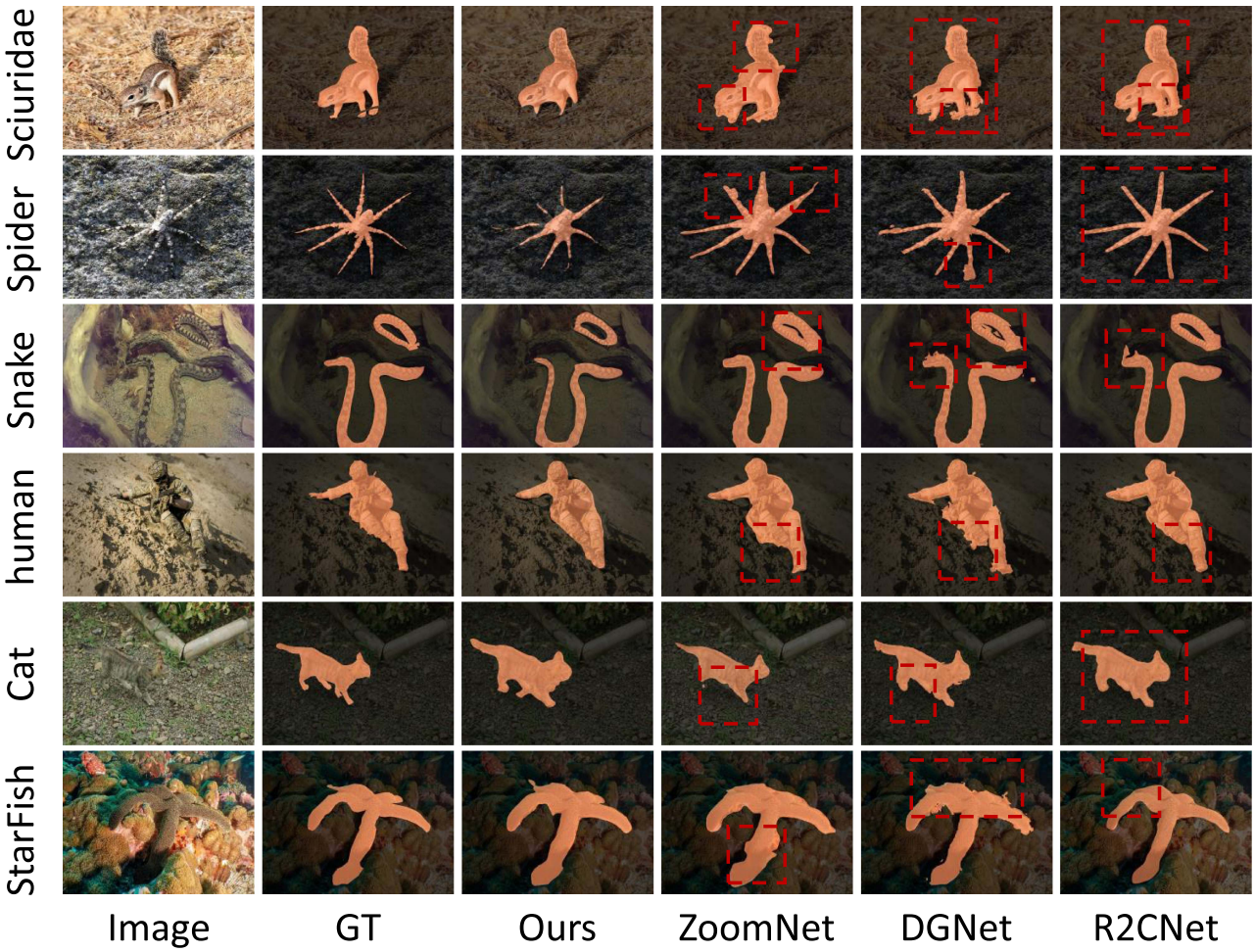

Fig. 5. Visual comparison of the segmentation results predicted by our model and other models. The segmentation masks are shown in orange.

TABLE II
QUANTITATIVE COMPARISON WITH OTHER METHODS, WHERE "↑" INDICATE THAT HIGHER VALUES ARE BETTER, WHEREAS "↓" INDICATE THAT LOWER VALUES ARE PREFERABLE; THE TOP TWO RESULTS OF EACH METRIC ARE HIGHLIGHTED IN **RED** AND **BLUE**.

Method	Origin	CAMO				NC4K			
		$M\downarrow$	$S_m\uparrow$	$wF\uparrow$	$\alpha E\uparrow$	$M\downarrow$	$S_m\uparrow$	$wF\uparrow$	$\alpha E\uparrow$
BGNet † [22]	IJCAI'22	0.073	0.812	0.749	0.870	0.044	0.851	0.788	0.907
ZoomNet † [23]	CVPR'22	0.066	0.820	0.752	0.878	0.043	0.853	0.784	0.896
PENet † [24]	IJCAI'23	0.063	0.828	0.771	0.890	0.042	0.855	0.795	0.912
FEDER † [26]	CVPR'23	0.066	0.836	0.807	0.897	0.042	0.862	0.824	0.913
PopNet † [27]	ICCV'23	0.077	0.808	0.744	0.859	0.042	0.861	0.802	0.909
CamoFocus † [18]	WACV'24	0.043	0.873	0.842	0.926	0.030	**0.899**	0.853	0.936
HGINet † [11]	TIP'24	0.041	**0.874**	0.848	0.937	0.027	**0.894**	**0.865**	0.947
VSCode † [10]	CVPR'24	0.060	0.836	0.818	0.915	0.038	0.874	0.853	0.930
CamoFormer † [9]	TPAMI'24	0.046	0.876	0.831	0.935	0.031	0.888	0.847	0.941
MCRNet † [29]	ArXiv'2025	0.054	0.854	**0.847**	0.915	0.036	0.875	0.857	0.930
Ours	TII'2025	**0.038**	**0.885**	**0.840**	**0.940**	**0.026**	0.889	**0.874**	**0.951**

recent, representative, and open-source camouflaged object segmentation methods on both the CAMO and NC4K benchmarks. Specifically, we include BGNet [22], ZoomNet [23], PENet [25], FEDER [26], PopNet [27], CamoFocus [18], HGINet [11], VSCode [10], and MCRNet [28]. Table II summarizes the quantitative results in terms of M, S_m, αE, wF, clearly demonstrating that our model achieves the lowest M and the highest wF on both datasets, thereby outperforming all competitors. On the CAMO dataset, our S_m index reaches 0.885, which is 1.3% higher than the second-best HGINet, indicating that our model is more robust in capturing global structures. The only difference is that it lags slightly behind MCRNet in wF, which may be because there is still room for improvement in detail fitting on extremely low-contrast samples, but overall, the gap is acceptable and does not affect the global performance advantage. On the NC4K dataset, our method is the best in all three indicators. Advanced methods such as CamoFormer and VSCode use fixed square windows or global attention of standard Transformer, while our method projects each window into an arbitrary quadrilateral, which can flexibly adapt to camouflaged targets of different directions and shapes, improving the alignment and expressiveness of feature extraction. In addition, their receptive field size is fixed, while our method can dynamically weight different receptive fields through decomposed large kernel convolution and spatial attention, and can adjust the context range for small and large targets, respectively, reducing misjudgments caused by too large or too small receptive fields.

3) Comparison With SOTA Methods on R2C7K Dataset With Single and Multiple Camouflaged Objects: In real-world scenarios, camouflaged objects may exist either as single targets or in the form of multiple targets. To comprehensively evaluate the performance of our model, we further divided all scenes in the R2C7K dataset into two groups: scenes with a single camouflaged object and scenes with multiple camouflaged objects. In single camouflaged object scenes $M = 0.032$, $S_m = 0.839$, $\alpha E = 0.900$, and $wF = 0.723$, whereas in multiple camouflaged object scenes $M = 0.058$, $S_m = 0.677$, $\alpha E = 0.835$, and $wF = 0.478$. As shown in Fig. 6, our model outperforms other SOTA methods in both scenarios.

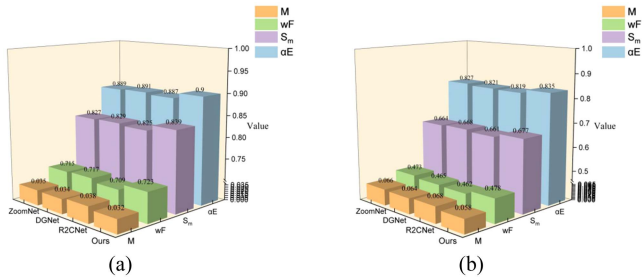

Fig. 6. Quantitative results comparing the proposed method with SOTA methods on the R2C7K dataset for single and multiple camouflaged objects. (a) Results for images containing only one camouflaged object. (b) Results for images containing multiple camouflaged objects.

4) Comparison With SOTA Methods on Different Categories of Comouflaged Targets in the COD10 K Dataset: We divided the COD10 K dataset into three categories: terrestrial animals, marine animals, and aerial animals, naming them COD-Terrestrial, COD-Aquatic, and COD-Flying, respectively, and conducted comparative experiments with SOTA methods. Through segmented evaluation, we can detect whether the same algorithm performs evenly in different fields (land/sea/air), which makes it easier to intuitively find situations where it is overfitting in a certain environment or weaker than in another ecological environment. As shown in Table III, for the terrestrial animal category, our model achieves metrics of $M = 0.027$, $S_m = 0.83$, $\alpha E = 0.906$, and $wF = 0.712$. Similarly, as presented in Table IV, for the marine animal category, the metrics are $M = 0.062$, $S_m = 0.786$, $\alpha E = 0.86$, and $wF = 0.664$. Furthermore, as illustrated in Table V, for the aerial animal category, the metrics are $M = 0.029$, $S_m = 0.849$, $\alpha E = 0.914$, and $wF = 0.735$. The results demonstrate that our model achieves outstanding segmentation performance and outperforms other SOTA methods.

TABLE III
QUANTITATIVE RESULTS COMPARING SOTA METHODS ON THE COD-TERRESTRIAL DATASET

Method	$M \downarrow$	$S_m \uparrow$	$\alpha E \uparrow$	$wF \uparrow$
ZoomNet [23]	0.031	0.827	0.894	0.698
DGNet [24]	0.030	0.824	0.900	0.701
R2CNet [19]	0.031	0.825	0.904	0.704
Ours	**0.027**	**0.830**	**0.906**	**0.712**

The bold values represent the best performance.

TABLE IV
QUANTITATIVE RESULTS COMPARING SOTA METHODS ON THE COD-AQUATIC DATASET

Method	$M \downarrow$	$S_m \uparrow$	$\alpha E \uparrow$	$wF \uparrow$
ZoomNet [23]	0.065	0.779	0.855	0.661
DGNEt [24]	0.067	0.781	0.851	0.657
R2CNet [19]	0.064	0.774	0.854	0.653
Ours	**0.062**	**0.786**	**0.860**	**0.664**

The bold values represent the best performance.

TABLE V
QUANTITATIVE RESULTS COMPARING SOTA METHODS ON THE COD-FLYING DATASET

Method	$M \downarrow$	$S_m \uparrow$	$\alpha E \uparrow$	$wF \uparrow$
ZoomNet [23]	0.032	0.826	0.905	0.720
DGNet [24]	0.031	0.829	0.899	0.729
R2Net [19]	0.034	0.821	0.900	0.715
Ours	**0.029**	**0.849**	**0.914**	**0.735**

The bold values represent the best performance.

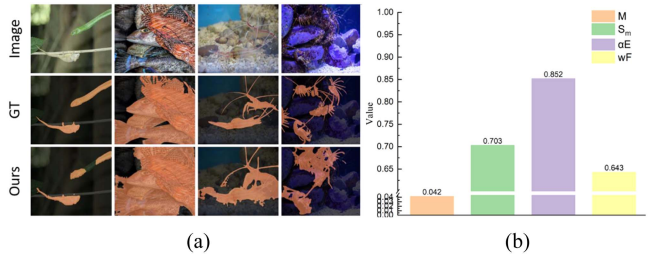

Fig. 7. (a) Segmentation visualization results. (b) Quantitative experimental metrics.

5) Results of the Multiclass Camouflage Target Experiment: We selected images containing multiple classes from the NC4K dataset for the experiment. Since the specific classes present in the images were unknown in advance, no reference image was provided. The metrics and visualization results are shown in Fig. 7.

6) Ablation Experiments and Sensitivity Analysis: Model modules: We conducted ablation experiments on the model modules using the COD10 K and R2C7K datasets, including the DWIM, the FEP module, and the receptive field adaptive adjustment (RFAA) module. For each dataset, we performed eight groups of experiments, including cases where none of the modules were applied, cases where one or two modules were applied, and cases where all modules were applied. As shown in Tables VI and VII, removing any of the aforementioned modules led to performance degradation. We believe that removing the DWIM restricts the model to fixed-size window settings, making it unable to flexibly capture features of different directions and shapes. Removing the feature perception module and the adaptive receptive field adjustment module results in the loss of the ability to capture sufficient contextual information and to adjust the receptive field according to the size of the camouflaged objects. These observations further demonstrate the effectiveness of the proposed mechanisms and module designs.

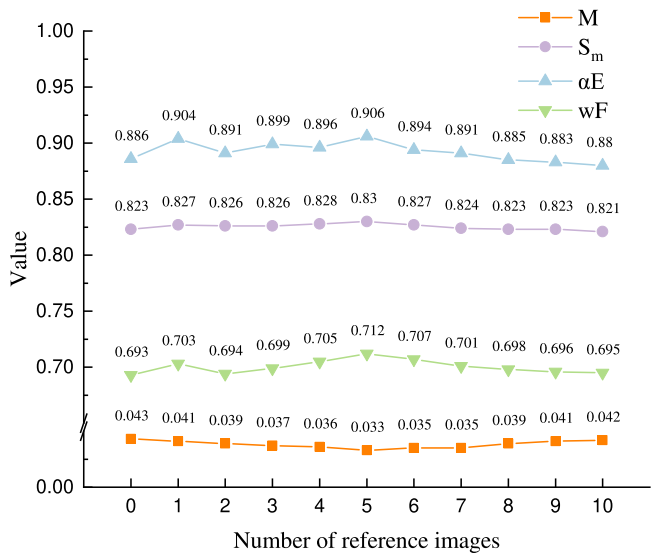

Fig. 8. Quantitative results when the number of reference images varies from 0 to 10.

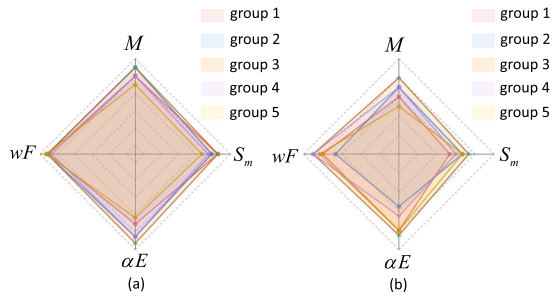

Fig. 9. Experimental results of different groups. Groups 1 to 5 represent different experimental settings.

TABLE VI
ABLATION EXPERIMENTS ON THE COMPONENTS OF OUR NETWORK ON THE COD10 K DATASET; "DWIM": DEFORMABLE WINDOW INTERACTION MECHANISM, "FEP": FEATURE ENHANCEMENT PERCEPTION MODULE, "RFAA": RECEPTIVE FIELD ADAPTIVE ADJUSTMENT MODULE.

No.	DWIM	FEP	RFAA	$M \downarrow$	$S_m \uparrow$	$\alpha E \uparrow$	$wF \uparrow$
1				0.037	0.810	0.881	0.687
2	✓			0.034	0.818	0.889	0.707
3		✓		0.036	0.815	0.887	0.699
4			✓	0.036	0.820	0.891	0.695
5	✓	✓		0.032	0.834	0.905	0.717
6	✓		✓	0.031	0.841	0.907	0.725
7		✓	✓	0.033	0.831	0.903	0.720
8	✓	✓	✓	**0.029**	**0.847**	**0.912**	**0.731**

The bold values represent the best performance.

TABLE VII
ABLATION EXPERIMENTS ON THE COMPONENTS OF OUR NETWORK ON THE R2C7K DATASET; "DWIM": DEFORMABLE WINDOW INTERACTION MECHANISM, "FEP": FEATURE ENHANCEMENT PERCEPTION MODULE, "RFAA": RECEPTIVE FIELD ADAPTIVE ADJUSTMENT MODULE.

No.	DWIM	FEP	RFAA	$M \downarrow$	$S_m \uparrow$	$\alpha E \uparrow$	$wF \uparrow$
1				0.039	0.804	0.870	0.664
2	✓			0.035	0.817	0.886	0.687
3		✓		0.033	0.829	0.891	0.700
4			✓	0.037	0.821	0.883	0.690
5	✓	✓		0.033	0.828	0.891	0.701
6	✓		✓	0.034	0.830	0.890	0.704
7		✓	✓	0.032	0.830	0.892	0.703
8	✓	✓	✓	**0.032**	**0.831**	**0.894**	**0.707**

The bold values represent the best performance.

Quantity of reference images: As shown in Fig. 8, we conducted ablation experiments on the COD-Terrestrial dataset to evaluate the performance with the number of reference images ranging from 0 to 10. We observed that the performance is optimal when the number of reference images is set to 5, while both excessive and insufficient numbers lead to performance degradation. We believe this is because an adequate number of reference images helps accurately locate the camouflaged objects, resulting in a more general representation. However, too many reference images may mislead the model.

Sensitivity of the loss function: We conducted experiments on the COD10 K and R2C7K datasets by assigning different weights to α and β in (17). The first group was set to $\alpha = 0.2$, $\beta = 0.8$, the second group to $\alpha = 0.4$, $\beta = 0.6$, the third group to $\alpha = 0.5$, $\beta = 0.5$, the fourth group to $\alpha = 0.6$, $\beta = 0.4$, and the fifth group to $\alpha = 0.8$, $\beta = 0.2$. As shown in Fig. 9, the performance of our model is both excellent and stable.

V. CONCLUSION

In this article, we proposed a novel method for applications in quality inspection in industrial manufacturing, species conservation, and safety monitoring. We utilized images containing salient targets as additional information to assist the model in detecting camouflaged objects hidden in the environment. To flexibly capture diverse multiscale features of camouflaged objects, we ingeniously incorporated the DWIM, the FEP module, and the RFAA module into the model. Compared with other SOTA methods on the COD10 K and R2C7K datasets, our method achieved state-of-the-art performance across four evaluation metrics. Additionally, we conducted comprehensive ablation experiments and sensitivity analyses, further verifying the effectiveness and robustness of our proposed model. However, we also observed that the model's performance decreases when multiple camouflaged objects are present in the scene. Addressing this issue will be the focus of our future research.

REFERENCES

[1] H. Bi, C. Zhang, K. Wang, J. Tong, and F. Zheng, "Rethinking camouflaged object detection: Models and datasets," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 9, pp. 5708–5724, Sep. 2022.

[2] D.-P. Fan, G.-P. Ji, G. Sun, M.-M. Cheng, J. Shen, and L. Shao, "Camouflaged object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2777–2787.

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

[4] Y. Lyu, H. Zhang, Y. Li, H. Liu, Y. Yang, and D. Yuan, "UEDG: Uncertainty-edge dual guided camouflage object detection," *IEEE Trans. Multimedia*, vol. 26, pp. 4050–4060, 2023.

[5] A. Li, J. Zhang, Y. Lv, B. Liu, T. Zhang, and Y. Dai, "Uncertainty-aware joint salient object and camouflaged object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 10071–10081.

[6] D.-P. Fan, G.-P. Ji, M.-M. Cheng, and L. Shao, "Concealed object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 6024–6042, Oct. 2022.

[7] Y. Pang, X. Zhao, T.-Z. Xiang, L. Zhang, and H. Lu, "Zoom in and out: A mixed-scale triplet network for camouflaged object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 2160–2170.

[8] D.-P. Fan, G.-P. Ji, G. Sun, M.-M. Cheng, J. Shen, and L. Shao, "Camouflaged object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 2774–2784.

[9] B. Yin et al., "CamoFormer: Masked separable attention for camouflaged object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 12, pp. 10362–10374, Dec. 2024.

[10] Z. Luo et al., "VSCode: General visual salient and camouflaged object detection with 2D prompt learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 17169–17180.

[11] S. Yao, H. Sun, T.-Z. Xiang, X. Wang, and X. Cao, "Hierarchical graph interaction transformer with dynamic token clustering for camouflaged object detection," *IEEE Trans. Image Process.*, vol. 33, pp. 5936–5948, 2024.

[12] L. Wang, H. Lu, X. Ruan, and M.-H. Yang, "Deep networks for saliency detection via local estimation and global search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3183–3192.

[13] H. Mei, G.-P. Ji, Z. Wei, X. Yang, X. Wei, and D.-P. Fan, "Camouflaged object segmentation with distraction mining," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8772–8781.

[14] Z. Zhang, W. Jin, J. Xu, and M.-M. Cheng, "Gradient-induced co-saliency detection," in *Proc. Comput. Vis.–ECCV: 16th Eur. Conf., Glasgow, U.K. Aug. 23–28, 2020*, 2020, pp. 455–472.

[15] Y. Piao, W. Ji, J. Li, M. Zhang, and H. Lu, "Depth-induced multi-scale recurrent attention network for saliency detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 7254–7263.

[16] J. Pan et al., "SalGAN: Visual saliency prediction with generative adversarial networks," 2017, *arXiv:1701.01081*.

[17] H. Lamdouar, C. Yang, W. Xie, and A. Zisserman, "Betrayed by motion: Camouflaged object discovery via motion segmentation," in *Proc. Asian Conf. Comput. Vis.*, 2020, pp 488–503.

[18] A. Khan, M. Khan, W. Gueaieb, A. El Saddik, G. De Masi, and F. Karray, "Camofocus: Enhancing camouflage object detection with split-feature focal modulation and context refinement," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2024, pp. 1434–1443.

[19] X. Zhang, B. Yin, Z. Lin, Q. Hou, D.-P. Fan, and M.-M. Cheng, "Referring camouflaged object detection," 2023, *arXiv:2306.07532*.

[20] M. Zhang, S. Xu, Y. Piao, D. Shi, S. Lin, and H. Lu, "PreyNet: Preying on camouflaged objects," in *Proc. 30th ACM Int. Conf. Multimedia*, 2022, pp. 5323–5332.

[21] H. Zhu et al., "I can find you! Boundary-guided separated attention network for camouflaged object detection," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 3608–3616.

[22] Y. Sun, S. Wang, C. Chen, and T.-Z. Xiang, "Boundary-guided camouflaged object detection," 2022, *arXiv:2207.00794*.

[23] Y. Pang, X. Zhao, T.-Z. Xiang, L. Zhang, and H. Lu, "Zoom in and out: A mixed-scale triplet network for camouflaged object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 2160–2170.

[24] 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," *Mach. Intell. Res.*, vol. 20, no. 1, pp. 92–108, 2023.

[25] Z. Huang et al., "Feature shrinkage pyramid for camouflaged object detection with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 5557–5566.

[26] C. He et al., "Camouflaged object detection with feature decomposition and edge reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. pattern Recognit.*, 2023, pp. 22046–22055.

[27] Z. Wu et al., "Source-free depth for object pop-out," in *Proc. IEEE/CVF Int. Conf. on Comput. Vis.*, 2023, pp. 1032–1042.

[28] D. Zhang, L. Cheng, Y. Liu, X. Wang, and J. Han, "Mamba capsule routing towards part-whole relational camouflaged object detection," 2025, *arXiv:2410.03987*.

Ruyu Liu (Member, IEEE) received the Ph.D. degree in control science and engineering from the Zhejiang University of Technology, Hangzhou, China, in 2021.

She currently works with the School of Information Science and Technology, Hangzhou Normal University, Hangzhou. Her research interests include 3-D vision and simultaneous localization and mapping (SLAM).

Guang Yang received the B.Eng. degree in computer science and technology from the Harbin University of Science and Technology, Harbin, China in 2022. He is currently working toward the M.Eng. degree in computer technology with the Tianjin University of Technology, Tianjin, China.

His current research interests include deep learning, 3-D vision, reinforcement learning, and machine vision.

Feng Xiao (Student Member, IEEE) received the M.S.E.E. degree in computer technology from the Tianjin University of Technology, Tianjin, China, in 2023. He is currently working toward the Ph.D. degree in marine engineering with the Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology, Trondheim, Norway.

His current research interests include computer vision, 3-D vision, and machine vision.

Jianhua Zhang (Senior Member, IEEE) received the Ph.D. degree in computer science from the University of Hamburg, Hamburg, Germany, in 2012.

He is currently a Professor with the School of Computer Science and Engineering, Tianjin University of Technology, Tianjin, China. His current research interests include SLAM, 3-D vision, reinforcement learning, and machine vision.

Shengyong Chen (Senior Member, IEEE) received the Ph.D. degree in computer vision from the City University of Hong Kong, Hong Kong, in 2003.

He is currently a Professor with the Tianjin University of Technology, Tianjin, China. He received a fellowship from the Alexander von Humboldt Foundation of Germany and worked with the University of Hamburg during 2006–2007. His research interests include computer vision, robotics, and image analysis.

Dr. Chen is a Fellow of IET and a Senior Member of CCF. He was the recipient of the National Outstanding Youth Foundation Award of China in 2013.