MambaCOD: Cross-modal Mamba Fusion Network with Adapter Tuning for RGB-D Camouflaged Object Detection

Jiesheng Wu[1][0000−0002−6941−3300]*, Lizheng Zhang[1]*, Fuyu Zhang[1], Yong Wu[1], Biao Jie[1]†, Hongchao Li[1], and Ji Du[2]

[1] School of Computer and Information, Anhui Normal University, Wuhu, 241002, China
jasonwu@mail.nankai.edu.cn, {zlz,zfy}@ahnu.edu.cn, wuyong_tj@163.com, jbiao@ahnu.edu.cn, lhc950304@foxmail.com,
[2] College of Artificial Intelligence, Nankai University, Tianjin, 300350, China
1120230244@mail.nankai.edu.cn

Abstract. Camouflaged object detection (COD) aims to identify objects that are visually blended into their surroundings. While depth maps are often used to enhance spatial understanding and generalization for COD, existing methods struggle with poor depth quality, limited feature extraction due to reliance on convolution or Transformer backbones, and inefficient cross-modal fusion with high computational cost. To address these issues, we propose MambaCOD, a novel RGB-D COD framework based on a Cross-modal Mamba Fusion Network with Adapter Tuning. Specifically, we first introduce the Camouflaged Cognitive Visual Adapter (Cona), which works with a frozen dual-stream VMamba backbone to extract effective RGB and depth features while preserving pretrained knowledge. Second, we design a Cross State Space Model (Cross-SSM) module that integrates a well-designed Shell-Like Scan (SLS) strategy and a Dual-SSM structure for efficient cross-modal fusion. Finally, an Edge Extraction Module (EEM) and a Decoder are incorporated to enhance edge awareness and multi-scale prediction. Extensive experiments on four benchmark datasets demonstrate that MambaCOD achieves state-of-the-art performance. Our codes will be available at: https://github.com/TomorrowJW/MambaCOD.

Keywords: Camouflaged object detection · Depth map · Adapter · Mamba.

1 Introduction

Camouflaged Object Detection (COD) aims to segment objects that are visually similar to and seamlessly embedded within their surroundings. COD has garnered increasing attention and found broad applications in diverse fields,

* Equal contribution.
† Corresponding author.

including species discovery [8], polyp segmentation [7], defect detection [1], autonomous driving [25], and so on.

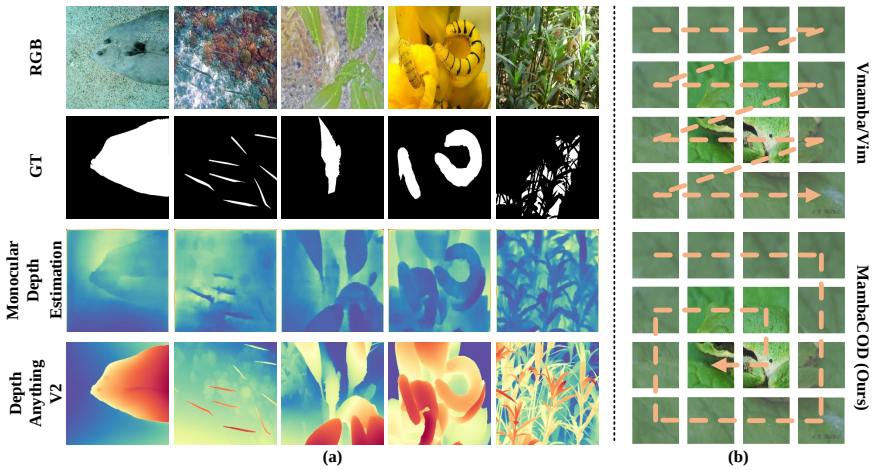

Fig. 1: **(a)** Depth maps from **Depth Anything V2** show superior quality compared to traditional depth estimators (3^{rd} row *vs.* 4^{th} row). **(b)** Unlike traditional Vision Mamba uses a Z-shaped scan that may disrupt spatial locality and continuity [16], our SLS uses a spiral strategy to enhance camouflaged object localization and preserve feature continuity.

Recently, while deep learning-based methods have achieved strong performance in COD, relying solely on 2D RGB features limits their effectiveness. To enhance detection, some works incorporate depth maps [29] and edge cues [26], but face notable challenges. First, Convolutional Neural Networks (CNNs) and Transformers used as backbones suffer from limited representation capacity and high computational cost, respectively. Second, low-quality pseudo-depth maps and simplistic fusion strategies hinder effective cross-modal fusion, leading to poor generalization and high overhead.

To address the above limitations in COD, we propose **MambaCOD**, *a Cross-modal Mamba Fusion Network with Adapter Tuning for RGB-D Camouflaged Object Detection*. Specifically, we first employ Depth Anything V2 [32] to generate high-quality pseudo-depth maps, as shown in Fig. 1(a). We then use a frozen dual-stream VMamba backbone [17] to extract RGB and depth features efficiently, leveraging its linear time complexity, efficient training-time parallelism, and strong capability in capturing key cues while filtering out irrelevant context. To further enhance camouflage-specific feature learning, we introduce **Cona**, a lightweight **C**amouflaged C**o**g**n**itive Visual **A**dapter, inspired by Mona [34], enabling Parameter-Efficient Fine-Tuning (PEFT). For cross-modal fusion, we design a **Cross State Space Model (Cross-SSM)** module that includes a

Shell-Like Scan (SLS) strategy and a Dual-SSM structure. Inspired by the idea of SARNet [31] that "Go Closer To See Better", and human visual focus patterns, SLS performs spiral scanning from center to periphery and vice versa (see Fig. 1(b)), improving object localization while maintaining semantic feature continuity. In contrast, traditional Z-shaped scan may disrupt locality and continuity [16]. The Dual-SSM enhances cross-modal interaction by performing parallel cross-calculations on RGB and depth features, enriched by global context to reinforce fusion quality. Finally, an **Edge Extraction Module (EEM)** and a **Decoder** are introduced to refine edge constraints and enhance multi-scale prediction. Our MambaCOD achieves State-Of-The-Art (SOTA) performance on four COD benchmarks. In a word, our main contributions are as follows:

- We propose **MambaCOD** for COD, which leverages Depth Anything V2 to generate high-quality depth maps to assist RGB-D COD, thereby enhancing detection performance.
- We propose a plug-and-play, PEFT adapter named **Cona**, which operates in conjunction with a frozen dual-stream VMamba backbone for effective RGB and depth feature extraction.
- We design a **Cross-SSM** module to facilitate efficient cross-modal fusion. It incorporates a **SLS** strategy to improve camouflaged object localization and a **Dual-SSM** mechanism to strengthen multi-modal representation.
- We develop an **EEM** and **a Decoder**, further enhancing edge-aware constraints and multi-scale feature representation to produce high-quality prediction maps.
- Extensive experiments on four benchmark datasets, i.e., CAMO, CHAMELEON, COD10K, and NC4K, demonstrate that our method achieves SOTA performance in both quantitative and qualitative evaluations.

2 Related Work

2.1 Camouflaged Object Detection

Recently, most researchers have mainly used deep learning methods for COD. Therefore, deep learning-based methods can be divided into three categories:

(1) **Feature learning-based methods**: These approaches typically design specialized architectures or modules (e.g., CNNs [6], Transformers [33]) to emulate biological hunting behaviors for accurate prediction. For instance, SINet [4] introduced a search-and-identify framework inspired by the human visual search mechanism; ZoomNet [23] leveraged multi-level contextual and global cues to enhance object localization; PFNet [22] adopted a perception-guided refinement strategy to iteratively improve prediction quality.
(2) **Knowledge-driven methods**: They usually employ the intrinsic knowledge of Multi-modal Large Language Models (MLLMs) to guide object recognition and segmentation. For example, Hu *et al.* [12] and Tang *et al.* [27] employed textual semantics to infer visual cues, thereby improving detection performance.

(3) **Prior knowledge- or auxiliary task-based methods**: These methods integrate extra cues such as edges [20,26], frequency components [37], depth information [29], and saliency maps [15] to further enhance generalization.

3 Method

3.1 Overall Architecture

The overall architecture of MambaCOD is illustrated in Fig. 2. RGB images and depth maps, where the depth maps are generated by the visual foundation model Depth Anything V2 [32], are fed into our model. MambaCOD employs a frozen dual-stream backbone enhanced with trainable Cona adapters, and incorporates a Cross-SSM module, an EEM, and a decoder. The backbones are used to extract RGB and depth features (denoted as $\boldsymbol{R_i}$ and $\boldsymbol{D_i}$, $i \in \{1,2,3,4,5\}$), which are then fused by the Cross-SSM to obtain enriched fusion representations (denoted as $\boldsymbol{F_i}$). Meanwhile, the EEM is employed to learn edge cues $\boldsymbol{F_e}$ to enhance edge constraints. Finally, the fusion features $\boldsymbol{F_i}$ and edge semantics $\boldsymbol{F_e}$ are combined in the decoder to generate the final prediction maps $\boldsymbol{P_j}$ ($j \in \{1,2,3,4\}$).

3.2 Camouflaged Cognitive Visual Adapter

To avoid catastrophic forgetting during fine-tuning [38], we adopt a PEFT strategy. Inspired by the excellent work named Mona [34], we design a COD-specific adapter named **Cona**, which enhances camouflaged feature representation while preserving the prior knowledge in the frozen VMamba. For clarity and convenience, we illustrate Cona using RGB features ($\boldsymbol{R_i}$) as an example.

As illustrated in Fig. 3, each RGB feature $\boldsymbol{R_i}$ undergoes LayerNorm and is scaled by a learnable factor $\boldsymbol{s_i}$, followed by element-wise addition and a linear layer for feature enhancement. To handle background similarity in COD, three scale-aware depth-wise separable convolutions (DW) with varying kernel sizes are applied and averaged. A lightweight residual block—comprising a 1×1 convolution, GELU, and DW—is then used. Finally, the output passes through GELU and a linear layer, with an original residual connection to produce $\boldsymbol{R_i^{'}}$. The whole process can be formulated as:

$$\boldsymbol{T^1_{r_i}} = \mathrm{Linear}(\mathrm{LN}(\boldsymbol{R_i}) \otimes \boldsymbol{s_1} \oplus \boldsymbol{R_i} \otimes \boldsymbol{s_2}), \tag{1}$$

$$\boldsymbol{T^2_{r_i}} = \mathrm{Avg}(\mathrm{DW}_3(\boldsymbol{T^1_{r_i}}) \otimes \boldsymbol{s_3} \oplus \mathrm{DW}_5(\boldsymbol{T^1_{r_i}}) \otimes \boldsymbol{s_4} \oplus \mathrm{DW}_7(\boldsymbol{T^1_{r_i}}) \otimes \boldsymbol{s_5} \oplus \boldsymbol{T^1_{r_i}}), \tag{2}$$

$$\boldsymbol{R_i^{'}} = \mathrm{Linear}(\sigma(\boldsymbol{T^2_{r_i}} \oplus \mathrm{DW}_3(\sigma(\mathrm{Conv}_{1\times1}(\boldsymbol{T^2_{r_i}})))) \oplus \boldsymbol{R_i}, \tag{3}$$

where $\mathrm{LN}(\cdot)$ denotes LayerNorm. $\mathrm{Linear}(\cdot)$ and $\mathrm{Avg}(\cdot)$ denote the linear layer and average function, respectively. $\boldsymbol{T^1_{r_i}}$ and $\boldsymbol{T^2_{r_i}}$ are intermediate variables. $\mathrm{DW}_n(\cdot)$ ($n \in \{3,5,7\}$) represents DW with kernel sizes 3, 5, and 7, respectively. The variables $\boldsymbol{s_i}$ ($i \in \{1,2,3,4,5\}$) are learnable scaling parameters, $\otimes$, $\oplus$ means element-wise multiplication and addition. $\sigma(\cdot)$ represent the GELU activation. $\mathrm{Conv}_{1\times1}(\cdot)$ stands for 1×1 convolution.

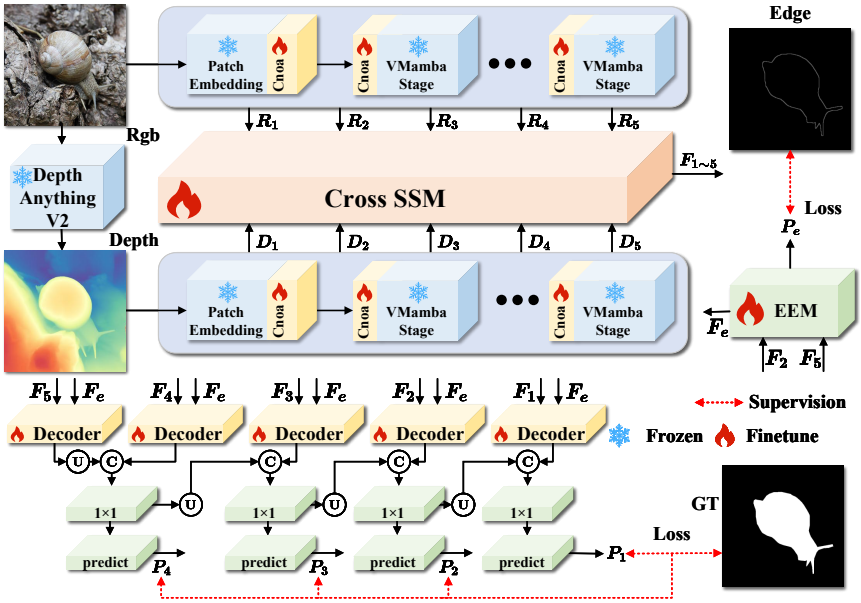

Fig. 2: The overall architecture of MambaCOD comprises four key stages. (1) High-quality depth maps are generated using Depth Anything V2. (2) A frozen dual-stream VMamba backbone, integrated with multiple Cona adapters, is employed for multi-modal feature extraction. (3) RGB and depth features are effectively fused through the Cross-SSM module. (4) A decoder, guided by the EEM, refines the features and outputs the final prediction maps.

3.3 Cross State Space Model

Basically, there exists a semantic gap between RGB and depth features, and direct fusion often introduces redundancy or conflicts, impairing overall performance. Although previous works [29] have attempted to address this, accurately distinguishing camouflaged objects remains challenging. Inspired by the State Space Model (SSM) in Mamba [9], we propose a Cross-SSM module to enable more effective cross-modal feature fusion. The structure of the Cross-SSM module is shown in Fig. 4(a). Given RGB features $\boldsymbol{R_i}$ and depth features $\boldsymbol{D_i}$, both are first processed by an Inner component. The outputs are then fed into the SLS component, followed by a Dual-SSM for cross-modal fusion (**detailed later**). Element-wise multiplication and residual connections refine the features, which are then passed through respective 1×1 CBR (Conv+Batch-Norm+RELU) layers. These features are fused via element-wise addition and softmax for refinement. Finally, further refinement is performed through a series of element-wise multiplication, addition, and concatenation operators, before being fed into an Outer component to produce the final fusion feature $\boldsymbol{F_i}$. The processes are de-

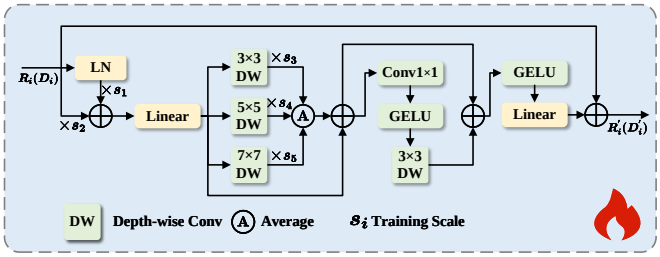

Fig. 3: Details of the fine-tuning adapter Cona.

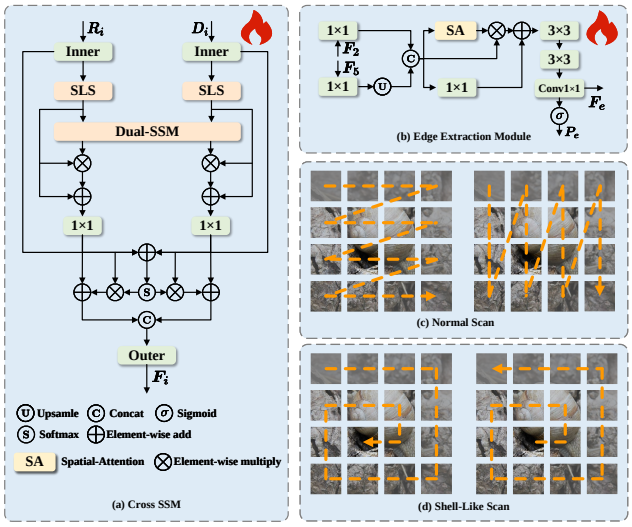

Fig. 4: (a) Architecture of the proposed Cross-SSM module. (b) Structure of the Edge Extraction Module (EEM). (c) Illustration of conventional scanning strategies. (d) Illustration of the proposed Shell-Like Scanning (SLS) strategies.

noted as follows:

$$\boldsymbol{R}_i^{'} = \text{Inner}(\boldsymbol{R}_i), \boldsymbol{D}_i^{'} = \text{Inner}(\boldsymbol{D}_i), \boldsymbol{R}_i^{t}, \boldsymbol{D}_i^{t} = \text{DS}(\text{SLS}(\boldsymbol{R}_i^{'}), \text{SLS}(\boldsymbol{D}_i^{'})), \tag{4}$$

$$\boldsymbol{R}_i^{''} = \text{CBR}_{1\times1}(\boldsymbol{R}_i^{t} \otimes \text{SLS}(\boldsymbol{R}_i^{'}) \oplus \text{SLS}(\boldsymbol{R}_i^{'})) \oplus \boldsymbol{R}_i^{'}, \tag{5}$$

$$\boldsymbol{D}_i^{''} = \text{CBR}_{1\times1}(\boldsymbol{D}_i^{t} \otimes \text{SLS}(\boldsymbol{D}_i^{'}) \oplus \text{SLS}(\boldsymbol{D}_i^{'})) \oplus \boldsymbol{D}_i^{'}, \tag{6}$$

$$\boldsymbol{F}_i^{t} = \text{S}(\boldsymbol{R}_i^{''} \oplus \boldsymbol{D}_i^{''}), \boldsymbol{F}_i = \text{Outer}(\text{Cat}((\boldsymbol{F}_i^{t} \otimes \boldsymbol{R}_i^{''} \oplus \boldsymbol{R}_i^{''}), (\boldsymbol{F}_i^{t} \otimes \boldsymbol{D}_i^{''} \oplus \boldsymbol{D}_i^{''}))), \tag{7}$$

where Inner$(\cdot)$ contains a $\text{CBR}_{1\times1}$ and a $\text{CBR}_{3\times3}$. $\boldsymbol{R}_i^{'}$ and $\boldsymbol{D}_i^{'}$ are the modality-specific features. $\text{CBR}_{n\times n}$ denotes a sequential combination of an $n \times n$ convolution, Batch Normalization (BN), and RELU activation. $\boldsymbol{R}_i^{''}$, $\boldsymbol{D}_i^{''}$, and $\boldsymbol{F}_i^{t}$ are intermediate variables. SLS$(\cdot)$ denotes the SLS component. DS$(\cdot)$ refers to the

Dual SSM module. S(·) means softmax. Cat(·) stands for concatenation. Outer(·) includes a $\text{CBR}_{3\times3}$ and a $\text{CBR}_{1\times1}$.

Shell-Like Scan In most COD scenarios, camouflaged objects often appear near the center of a scene. Inspired by the idea "Go Closer To See Better" of SARNet [31] and human visual focus patterns, we propose a novel Shell-Like Scan (SLS) strategy (Fig. 4(d)). Unlike traditional Z-shaped scans that operate row- or column-wise (see Fig. 4(c)), SLS adopts a spiral pattern centered on the image, enabling more effective camouflaged feature extraction in central regions while preserving semantic continuity.

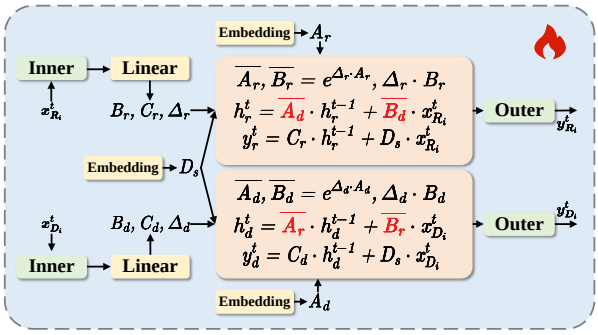

Fig. 5: Overall structures of Dual SSM

Dual SSM To enhance spatial semantic discrimination, we utilize depth as an auxiliary modality and propose the Dual SSM module for improved cross-modal fusion. Dual SSM captures rich global context and aids in identifying objects that are visually blended into the background. As shown in Fig. 5, the input RGB feature $\boldsymbol{x}^{\boldsymbol{t}}_{\boldsymbol{R_i}}$ and depth feature $\boldsymbol{x}^{\boldsymbol{t}}_{\boldsymbol{D_i}}$ are first processed by the Inner module. A linear layer then generates three modality-specific parameter sets: $\boldsymbol{B_{r/d}}$, $\boldsymbol{C_{r/d}}$, and $\boldsymbol{\Delta_{r/d}}$. Additionally, modality-specific parameters $\boldsymbol{A_{r/d}}$ and a shared parameter $\boldsymbol{D_s}$ are randomly initialized. The full formulation is as follows:

$$\boldsymbol{B_r}, \boldsymbol{C_r}, \boldsymbol{\Delta_r} = \text{Linear}(\text{Inner}(\boldsymbol{x}^{\boldsymbol{t}}_{\boldsymbol{R_i}})), \boldsymbol{B_d}, \boldsymbol{C_d}, \boldsymbol{\Delta_d} = \text{Linear}(\text{Inner}(\boldsymbol{x}^{\boldsymbol{t}}_{\boldsymbol{D_i}})). \quad (8)$$

Next, the depth parameters $\overline{\boldsymbol{A_d}}$ and $\overline{\boldsymbol{B_d}}$ are applied to the RGB features, while the RGB parameters $\overline{\boldsymbol{A_r}}$ and $\overline{\boldsymbol{B_r}}$ are applied to the depth features. The shared parameter D_s is used across both modalities. The formulations are as follows:

$$\overline{\boldsymbol{A_r}}, \overline{\boldsymbol{B_r}} = \boldsymbol{e}^{\boldsymbol{\Delta_r A_r}}, \boldsymbol{\Delta_r B_r}, \overline{\boldsymbol{A_d}}, \overline{\boldsymbol{B_d}} = \boldsymbol{e}^{\boldsymbol{\Delta_d A_d}}, \boldsymbol{\Delta_d B_d}, \quad (9)$$

$$\boldsymbol{h^t_r} = \overline{\boldsymbol{A_d}}\boldsymbol{h^{t-1}_r} + \overline{\boldsymbol{B_d}}\boldsymbol{x^t_{R_i}}, \boldsymbol{y^t_r} = \boldsymbol{C_r h^{t-1}_r} + \boldsymbol{D_s x^t_{R_i}}, \quad (10)$$

$$\boldsymbol{h^t_d} = \overline{\boldsymbol{A_r}}\boldsymbol{h^{t-1}_d} + \overline{\boldsymbol{B_r}}\boldsymbol{x^t_{D_i}}, \boldsymbol{y^t_d} = \boldsymbol{C_d h^{t-1}_d} + \boldsymbol{D_s x^t_{D_i}}. \quad (11)$$

Finally, the outputs $\boldsymbol{y}_{\boldsymbol{R}_i}^{\boldsymbol{t}}$ and $\boldsymbol{y}_{\boldsymbol{D}_i}^{\boldsymbol{t}}$ are fed into the Outer layer, as defined below:

$$\boldsymbol{y}_{\boldsymbol{R}_i}^{\boldsymbol{t}} = \text{Outer}(\boldsymbol{y}_{\boldsymbol{r}}^{\boldsymbol{t}}), \boldsymbol{y}_{\boldsymbol{D}_i}^{\boldsymbol{t}} = \text{Outer}(\boldsymbol{y}_{\boldsymbol{d}}^{\boldsymbol{t}}). \tag{12}$$

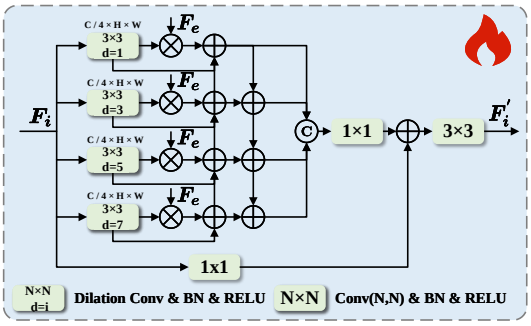

Fig. 6: Details of the decoder.

3.4 Edge Extraction Module

To enhance edge constraints and improve generalization, we propose an EEM to learn edge-aware features. As shown in Fig. 4(b), EEM takes fusion features $\boldsymbol{F_2}$ and $\boldsymbol{F_5}$ as inputs, following the design of BGNet [26]. These features are first processed by a $\text{CBR}_{1\times1}$ block and upsampled for alignment. The aligned features are concatenated into intermediate variables $\boldsymbol{F_c}$ and passed through a spatial attention (SA) module to capture spatial context. The attended features are then refined via element-wise multiplication with $\boldsymbol{F_c}$ and a residual connection through another $\text{CBR}_{1\times1}$ block. Finally, two convolutional layers generate the edge feature map $\boldsymbol{F_e}$ and its corresponding edge prediction $\boldsymbol{P_e}$, as formulated below:

$$\boldsymbol{F_c} = \text{Cat}(\text{CBR}_{1\times1}(\boldsymbol{F_2}), \text{Up}(\text{CBR}_{1\times1}(\boldsymbol{F_5}))), \tag{13}$$

$$\boldsymbol{F_e} = \text{Convs}((\text{SA}(\boldsymbol{F_c}) \otimes \boldsymbol{F_c}) \oplus \text{CBR}_{1\times1}(\boldsymbol{F_c})), \tag{14}$$

$$\boldsymbol{P_e} = \boldsymbol{S}(\boldsymbol{F_e}), \tag{15}$$

where $\text{Up}(\cdot)$ is upsampling operator. $\text{Convs}(\cdot)$ denotes two $\text{CBR}_{3\times3}$ blocks followed by a 1×1 convolution. $\boldsymbol{S}(\cdot)$ and $\text{SA}(\cdot)$ represent the Sigmoid function and Spatial Attention, respectively.

3.5 Decoder

As in Fig. 6(a), input features $\boldsymbol{F_i} \in \mathbb{R}^{C\times H\times W}$ pass through four parallel $\text{CBR}_{1\times1}$ blocks. The outputs are split into four groups, each processed by dilated convolutions with increasing dilation rates, producing intermediate variables $\boldsymbol{F_i^{k'}}$

($k = 1, 2, 3, 4$). Each $\boldsymbol{F_i^{k'}}$ is modulated by edge features $\boldsymbol{F_e}$ to enhance boundaries, yielding $\boldsymbol{F_i^k}$. Adjacent groups are fused via element-wise addition to enable inter-group interaction, then concatenated to form the unified feature $\boldsymbol{F_i^t}$. Finally, residual refinement combines $\boldsymbol{F_i^t}$ with the original input through $\mathrm{CBR}_{1\times1}$ and $\mathrm{CBR}_{3\times3}$ layers, producing the final output $\boldsymbol{F_i^{'}}$. The whole process is formulated as follows:

$$\boldsymbol{F_i^{k'}} = \mathrm{DC}_{2k-1}(\mathrm{CBR}_{1\times1}(\boldsymbol{F_i})), k \in \{1, 2, 3, 4\}, \tag{16}$$

$$\boldsymbol{F_i^k} = \boldsymbol{F_i^{k'}} \otimes \boldsymbol{F_e} \oplus \boldsymbol{F_i^{k'}}, k \in \{1, 2, 3, 4\}, \tag{17}$$

$$\boldsymbol{F_i^k} = \boldsymbol{F_i^{k-1}} \oplus \boldsymbol{F_i^k} \oplus \boldsymbol{F_i^{k+1}}, k \in \{1, 2, 3, 4\}, \tag{18}$$

$$\boldsymbol{F_i^t} = \mathrm{Cat}(\boldsymbol{F_i^1}, \boldsymbol{F_i^2}, \boldsymbol{F_i^3}, \boldsymbol{F_i^4}), \tag{19}$$

$$\boldsymbol{F_i^{'}} = \mathrm{CBR}_{3\times3}(\mathrm{CBR}_{1\times1}(\boldsymbol{F_i^t}) \oplus \mathrm{CBR}_{1\times1}(\boldsymbol{F_i})), \tag{20}$$

where $\mathrm{DC}_d(\cdot)$ denotes a $\mathrm{CBR}_{1\times1}$ layer with dilation rate d, and $\boldsymbol{F_e}$ represents the edge features from Section 3.4. Finally, $\boldsymbol{F_i^{'}}$ is input into a 1×1 convolution to obtain the desired output $\boldsymbol{P_j}$.

3.6 Loss Fuction

For edge supervision, we utilize the dice loss ($\mathcal{L}_{dice}$). For mask supervision, we adopt the structure loss ($\mathcal{L}_{struct}$), which combines weighted binary cross-entropy loss ($\mathcal{L}_{bce}^{w}$) and weighted IOU loss ($\mathcal{L}_{iou}^{w}$). The total loss functions are:

$$\mathcal{L}_{struct} = \mathcal{L}_{bce}^{w} + \mathcal{L}_{iou}^{w}, \mathcal{L}_{tol} = \sum_{j=1}^{4}(\mathcal{L}_{struct}(\boldsymbol{P_j})) + 4 \times \mathcal{L}_{dice}(\boldsymbol{P_e}), \tag{21}$$

where $\boldsymbol{P_j}$ ($j \in \{1, 2, 3, 4\}$) stands for predictions. $\boldsymbol{P_e}$ is the edge prediction.

4 Experiment

4.1 Datasets and Metrics

To verify the superiority of our method, we evaluate our MambaCOD on four widely used datasets: CAMO [14], CHAMELEON [24], COD10K [6], and NC4K [19]. Follow the prior settings [6], four metrics are used for evaluation: mean absolute error($\mathcal{M}$) [2], weighted F-measure(F_β^w) [21], S-measure(S_α) [3], mean E-measure(E_ϕ) [5].

4.2 Implementation Details

MambaCOD is implemented in PyTorch with a frozen VMamba backbone pretrained on ImageNet. Training uses 4,040 images from CAMO and COD10K, augmented by random flipping and rotation. The model is trained for 120 epochs with a batch size of 8 on two RTX 3090 GPUs. Inputs are resized to 416×416 and optimized using Adam (weight decay 1×10^{-4}, initial learning rate 5×10^{-4}, halved every 20 epochs).

Table 1: The quantitative comparison with SOTA methods on datasets CAMO, CHAMELEON, COD10K, and NC4K under four widely used evaluation metrics S_α, F_β^w, E_ϕ, and $\mathcal{M}$. The symbols ↑ and ↓ indicate that larger and smaller values are better, respectively. The best results are highlighted in **bold**.

Method	Publications	CAMO (250)				CHAMELEON (76)				COD10K (2,026)				NC4K (4,121)			
		$S_\alpha\uparrow$	$F_\beta^w\uparrow$	$E_\phi\uparrow$	$M\downarrow$	$S_\alpha\uparrow$	$F_\beta^w\uparrow$	$E_\phi\uparrow$	$M\downarrow$	$S_\alpha\uparrow$	$F_\beta^w\uparrow$	$E_\phi\uparrow$	$M\downarrow$	$S_\alpha\uparrow$	$F_\beta^w\uparrow$	$E_\phi\uparrow$	$M\downarrow$
SINetv2 [4]	TPAMI21	0.820	0.743	0.882	0.070	0.888	0.816	0.942	0.030	0.815	0.680	0.887	0.037	0.847	0.769	0.903	0.048
PFNet [22]	CVPR21	0.782	0.695	0.855	0.085	0.882	0.810	0.942	0.033	0.800	0.660	0.877	0.040	0.829	0.745	0.887	0.053
ZoomNet [23]	CVPR22	0.820	0.752	0.877	0.066	0.902	0.845	0.952	0.023	0.838	0.729	0.888	0.029	0.853	0.784	0.896	0.043
FEDER [11]	CVPR23	0.836	0.807	0.897	0.066	0.887	0.834	0.946	0.030	0.844	0.748	0.911	0.029	0.862	0.824	0.913	0.042
DaCOD [29]	ACMMM23	0.855	0.796	0.911	0.051	–	–	–	–	0.840	0.729	0.908	0.028	0.874	0.814	0.923	0.035
PopNet [30]	ICCV23	0.806	0.821	0.869	0.073	0.910	**0.893**	0.952	0.022	0.827	0.789	0.897	0.031	0.852	0.852	0.908	0.043
FSPNet [13]	CVPR23	0.851	0.802	0.905	0.056	0.908	0.868	0.943	0.022	0.850	0.755	0.912	0.028	0.879	0.816	0.914	0.035
RISNet [28]	CVPR24	0.870	0.827	0.922	0.050	0.908	0.863	0.951	0.024	0.873	0.799	0.931	0.025	0.882	0.834	0.926	0.037
ABNet [36]	PRCV24	0.836	0.766	0.888	0.066	–	–	–	–	0.836	0.715	0.905	0.032	0.859	0.793	0.912	0.041
VSCode [18]	CVPR24	0.873	0.820	0.925	0.046	–	–	–	–	0.869	0.780	0.931	0.023	0.891	0.841	0.929	0.032
FocusDiffuser-ViT [35]	ECCV24	0.869	0.842	0.931	0.043	–	–	–	–	0.863	0.785	**0.934**	0.024	0.882	0.840	0.933	0.032
SENet [10]	TIP25	0.888	0.847	0.932	0.039	0.918	0.878	**0.957**	0.019	0.865	0.780	0.925	0.024	0.889	0.843	0.933	0.032
MambaCOD(Ours)	**2025**	**0.890**	**0.853**	**0.934**	**0.039**	**0.921**	0.886	0.950	**0.019**	**0.883**	**0.813**	0.932	**0.020**	**0.897**	**0.853**	**0.935**	**0.030**

4.3 Comparison with SOTA

To demonstrate the superiority of our method, we compared 12 SOTA methods, including SINetv2 [4], PFNet [22], ZoomNet [23], FEDER [11], DaCOD [29], PopNet [30], FSPNet [13], RISNet [28], ABNet [36], VSCode [18], FocusDiffuser [35], and SENet [10].

Quantitative Comparison As shown in Table 1, our method achieves SOTA performance. On the four benchmark datasets, it brings average improvements of 0.88%, 1.76%, 0.11%, and 5.73% in S_α, F_β^w, E_ϕ, and $\mathcal{M}$, respectively. Notably, on the large-scale NC4K dataset, it achieves a 6.25% gain in $\mathcal{M}$. These results highlight the effectiveness and generalization ability of our model.

Qualitative Comparison We qualitatively compare our model with existing methods, as illustrated in Fig. 7. The results show that our model accurately detects camouflaged objects across diverse scenarios, including occlusion (1st row), large (rows 2, 5) and small objects (rows 3, 4), high similarity (rows 2–5), fine details (row 5), and multiple targets (row 6), where other methods often perform poorly in these scenarios. Overall, our model demonstrates superior generalization in challenging camouflaged scenes.

4.4 Ablation Study

The ablation experiments are performed to verify the effectiveness of our proposed model. The results are shown in Table 2. We remove all modules and leave only a dual-stream backbone as the baseline (denoted as "B").

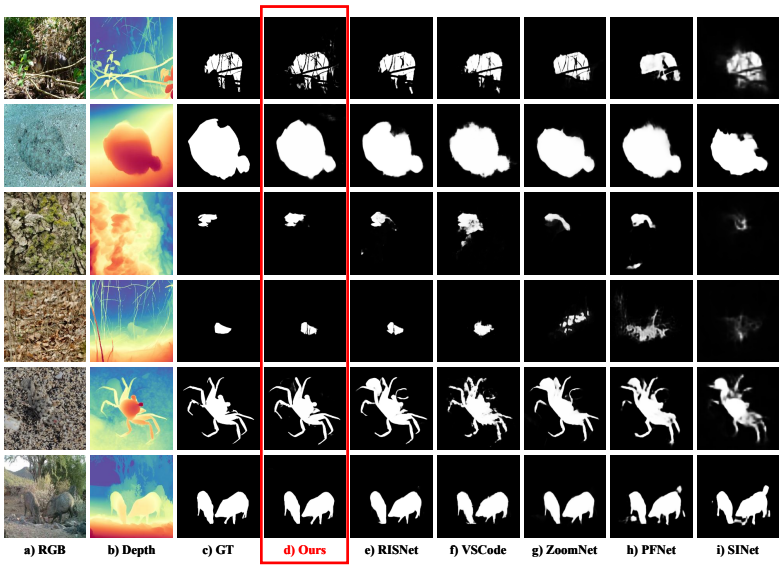

Fig. 7: Qualitative comparison of our model with other methods.

Effectiveness of Cross SSM We integrate the proposed module into the baseline "B", denoted as "B+C". The corresponding ablation results are presented in the "B+C" row of Table 2. The configuration achieves average improvements of 1.23%, 3.25%, 1.16%, and 12.73% across the four metrics. These results verify the effectiveness of the Cross-SSM in integrating RGB and depth features, substantially enhancing camouflaged object detection performance.

Effectiveness of Decoder Similarly, we further add "D" to "B+C", denoted as "B+C+D". As shown in the corresponding row of Table 2, this addition enables the model to capture rich multi-scale semantics, leading to performance gains.

Effectiveness of EEM To validate the effectiveness of EEM, we add module "E" to "B+C+D", forming the complete model "B+C+D+E". As shown in the corresponding row of Table 2, incorporating edge features further improves performance. Specifically, our model achieves average gains of 0.53%, 0.94%, 0.39%, and 3.50% across the four evaluations.

Effectiveness of Cona To evaluate the effectiveness of Cona fine-tuning, we compared it with a fully fine-tuned paradigm. Results in Table 2 show significant improvements, with average improvements of 3.94%, 7.37%, 3.67%, and 27.23% in terms of the four evaluations, respectively. The results demonstrate that freezing the backbone and embedding Cona enhances feature extraction while preserving prior knowledge.

Table 2: Ablation study of each module in our model. The top section corresponds to Full Fine-tuning, and the bottom to Cona Fine-tuning. "B" denotes the Backbone (baseline), "C" the Cross-SSM, "D" the Decoder, and "E" the EEM.

Method		CAMO (250)				COD10K (2,026)				NC4K (4,121)			
		$S_\alpha \uparrow$	$F_\beta^w \uparrow$	$E_\phi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta^w \uparrow$	$E_\phi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta^w \uparrow$	$E_\phi \uparrow$	$\mathcal{M} \downarrow$
Full Fine-tuning	B	0.855	0.796	0.894	0.053	0.834	0.717	0.885	0.032	0.861	0.792	0.901	0.044
	B+C	0.857	0.800	0.904	0.050	0.830	0.712	0.881	0.033	0.860	0.788	0.899	0.044
	B+C+D	0.851	0.791	0.899	0.054	0.839	0.731	0.888	0.030	0.862	0.796	0.900	0.043
	B+C+D+E	0.863	0.811	0.907	0.049	0.842	0.738	0.892	0.029	0.864	0.799	0.903	0.043
Cona Fine-tuning	B	0.877	0.827	0.922	0.044	0.856	0.755	0.913	0.026	0.879	0.818	0.922	0.035
	B+C	0.886	0.847	0.932	0.039	0.870	0.790	0.926	0.022	0.888	0.840	0.931	0.031
	B+C+D	0.888	0.850	0.930	0.040	0.875	0.799	0.928	0.021	0.893	0.847	0.932	0.031
	B+C+D+E(Ours)	**0.890**	**0.853**	**0.934**	**0.039**	**0.883**	**0.813**	**0.932**	**0.020**	**0.897**	**0.853**	**0.935**	**0.030**

Table 3: Comparisons of different scanning strategies.

Method	CAMO (250)				COD10K (2,026)				NC4K (4,121)			
	$S_\alpha \uparrow$	$F_\beta^w \uparrow$	$E_\phi \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$F_\beta^w \uparrow$	$E_\phi \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$F_\beta^w \uparrow$	$E_\phi \uparrow$	$M \downarrow$
No-Scan	0.886	0.839	0.925	0.044	0.879	0.797	0.927	0.022	0.898	0.848	0.934	0.031
Normal-Scan	0.890	0.850	0.933	0.040	0.880	0.801	0.930	0.021	**0.900**	0.853	0.934	0.031
Shell-Like-Scan (Ours)	**0.890**	**0.853**	**0.934**	**0.039**	**0.883**	**0.813**	**0.932**	**0.020**	0.897	**0.853**	**0.935**	**0.030**

Effectiveness of SLS We evaluate the effectiveness of our proposed SLS strategy, with results shown in Table Table 3. Two additional settings, no-scan strategy and normal-scan strategy, are tested for comparison. Our SLS strategy outperforms both, confirming its superior suitability for COD tasks.

5 Conclusion

In this work, we proposed MambaCOD, an efficient RGB-D COD network. It employs a frozen dual-stream backbone enhanced by the Cona adapter to boost feature extraction. The Cross-SSM module facilitates effective RGB-D fusion, while the SLS mechanism guides the model in identifying camouflaged objects within complex backgrounds. Additionally, the decoder, aided by edge-guided information, enables precise object localization. Extensive experiments on multiple benchmarks show that MambaCOD outperforms 12 state-of-the-art methods, validating its effectiveness. In future work, we plan to incorporate textual knowledge to further enhance RGB-D COD.

6 Acknowledgement

This work was supported by the National Natural Science Foundation of China (No.62302013) and partly supported by the Natural Science Foundation of Anhui Province (No.2308085QF220, No.2408085MF169).

References

1. Bhajantri, N.U., Nagabhushan, P.: Camouflage defect identification: a novel approach. In: 9th International Conference on Information Technology (ICIT'06). pp. 145–148. IEEE (2006)
2. Borji, A., Cheng, M.M., Jiang, H., Li, J.: Salient object detection: A benchmark. IEEE transactions on image processing **24**(12), 5706–5722 (2015)
3. Fan, D.P., Cheng, M.M., Liu, Y., Li, T., Borji, A.: Structure-measure: A new way to evaluate foreground maps. In: Proceedings of the IEEE international conference on computer vision. pp. 4548–4557 (2017)
4. Fan, D.P., Ji, G.P., Cheng, M.M., Shao, L.: Concealed object detection. IEEE TPAMI **44**(10), 6024–6042 (2022)
5. Fan, D.P., Ji, G.P., Qin, X., Cheng, M.M.: Cognitive vision inspired object segmentation metric and loss function. Scientia Sinica Informationis **6**(6), 5 (2021)
6. Fan, D.P., Ji, G.P., Sun, G., Cheng, M.M., Shen, J., Ling, S.: Camouflaged object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
7. Fan, D.P., Ji, G.P., Zhou, T., Chen, G., Fu, H., Shen, J., Shao, L.: Pranet: Parallel reverse attention network for polyp segmentation. In: International conference on medical image computing and computer-assisted intervention. pp. 263–273. Springer (2020)
8. Pérez-de la Fuente, R., Delclòs, X., Peñalver, E., Speranza, M., Wierzchos, J., Ascaso, C., Engel, M.S.: Early evolution and ecology of camouflage in insects. Proceedings of the National Academy of Sciences **109**(52), 21414–21419 (2012)
9. Gu, A., Dao, T.: Mamba: Linear-time sequence modeling with selective state spaces. arXiv preprint arXiv:2312.00752 (2023)
10. Hao, C., Yu, Z., Liu, X., Xu, J., Yue, H., Yang, J.: A simple yet effective network based on vision transformer for camouflaged object and salient object detection. IEEE Transactions on Image Processing **34**, 608–622 (2025)
11. He, C., Li, K., Zhang, Y., Tang, L., Zhang, Y., Guo, Z., Li, X.: Camouflaged object detection with feature decomposition and edge reconstruction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 22046–22055 (2023)
12. Hu, J., Lin, J., Gong, S., Cai, W.: Relax image-specific prompt requirement in sam: A single generic prompt for segmenting camouflaged objects. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 12511–12518 (2024)
13. Huang, Z., Dai, H., Xiang, T.Z., Wang, S., Chen, H.X., Qin, J., Xiong, H.: Feature shrinkage pyramid for camouflaged object detection with transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5557–5566 (2023)
14. Le, T.N., Nguyen, T.V., Nie, Z., Tran, M.T., Sugimoto, A.: Anabranch network for camouflaged object segmentation. Journal of Computer Vision and Image Understanding **184**, 45–56 (2019)
15. Li, A., Zhang, J., Lv, Y., Liu, B., Zhang, T., Dai, Y.: Uncertainty-aware joint salient object and camouflaged object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10071–10081 (2021)
16. Li, B., Zhao, H., Wang, W., Hu, P., Gou, Y., Peng, X.: Mair: A locality-and continuity-preserving mamba for image restoration. arXiv preprint arXiv:2412.20066 (2024)

17. Liu, Y., Tian, Y., Zhao, Y., Yu, H., Xie, L., Wang, Y., Ye, Q., Jiao, J., Liu, Y.: Vmamba: Visual state space model. Advances in neural information processing systems **37**, 103031–103063 (2024)
18. Luo, Z., Liu, N., Zhao, W., Yang, X., Zhang, D., Fan, D.P., Khan, F., Han, J.: Vscode: General visual salient and camouflaged object detection with 2d prompt learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 17169–17180 (June 2024)
19. Lv, Y., Zhang, J., Dai, Y., Li, A., Liu, B., Barnes, N., Fan, D.P.: Simultaneously localize, segment and rank the camouflaged objects. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11586–11596 (2021)
20. Lyu, Y., Zhang, H., Li, Y., Liu, H., Yang, Y., Yuan, D.: Uedg: Uncertainty-edge dual guided camouflage object detection. IEEE Transactions on Multimedia **26**, 4050–4060 (2024)
21. Margolin, R., Zelnik-Manor, L., Tal, A.: How to evaluate foreground maps? In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 248–255 (2014)
22. Mei, H., Ji, G.P., Wei, Z., Yang, X., Wei, X., Fan, D.P.: Camouflaged object segmentation with distraction mining. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8772–8781 (2021)
23. Pang, Y., Zhao, X., Xiang, T.Z., Zhang, L., Lu, H.: Zoom in and out: A mixed-scale triplet network for camouflaged object detection. In: Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition. pp. 2160–2170 (2022)
24. Skurowski, P., Abdulameer, H., Błaszczyk, J., Depta, T., Kornacki, A., Kozieł, P.: Animal camouflage analysis: Chameleon database. Unpublished manuscript **2**(6), 7 (2018)
25. Sun, B., Ma, M., Yuan, N., Li, J., Yu, T.: Detecting the background-similar objects in complex transportation scenes. IEEE Transactions on Intelligent Transportation Systems **25**(3), 2920–2932 (2023)
26. Sun, Y., Wang, S., Chen, C., Xiang, T.Z.: Boundary-guided camouflaged object detection. arXiv preprint arXiv:2207.00794 (2022)
27. Tang, L., Jiang, P.T., Shen, Z.H., Zhang, H., Chen, J.W., Li, B.: Chain of visual perception: Harnessing multimodal large language models for zero-shot camouflaged object detection. In: Proceedings of the 32nd ACM International Conference on Multimedia. pp. 8805–8814 (2024)
28. Wang, L., Yang, J., Zhang, Y., Wang, F., Zheng, F.: Depth-aware concealed crop detection in dense agricultural scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17201–17211 (2024)
29. Wang, Q., Yang, J., Yu, X., Wang, F., Chen, P., Zheng, F.: Depth-aided camouflaged object detection. In: Proceedings of the 31st ACM International Conference on Multimedia. MM '23, Association for Computing Machinery (2023)
30. Wu, Z., Paudel, D.P., Fan, D.P., Wang, J., Wang, S., Demonceaux, C., Timofte, R., Van Gool, L.: Source-free depth for object pop-out. In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 1032–1042 (2023)
31. Xing, H., Gao, S., Wang, Y., Wei, X., Tang, H., Zhang, W.: Go closer to see better: Camouflaged object detection via object area amplification and figure-ground conversion. IEEE Transactions on Circuits and Systems for Video Technology **33**(10), 5444–5457 (2023)
32. Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., Zhao, H.: Depth anything v2. Advances in Neural Information Processing Systems **37**, 21875–21911 (2024)

33. Yin, B., Zhang, X., Fan, D.P., Jiao, S., Cheng, M.M., Van Gool, L., Hou, Q.: Camoformer: Masked separable attention for camouflaged object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024)
34. Yin, D., Hu, L., Li, B., Zhang, Y., Yang, X.: 5%¿ 100%: Breaking performance shackles of full fine-tuning on visual recognition tasks. arXiv preprint arXiv:2408.08345 (2024)
35. Zhao, J., Li, X., Yang, F., Zhai, Q., Luo, A., Jiao, Z., Cheng, H.: Focusdiffuser: Perceiving local disparities for camouflaged object detection. In: European Conference on Computer Vision. pp. 181–198. Springer (2024)
36. Zhong, J., Wang, A.: Attention and boundary induced feature refinement network for camouflaged object detection. In: Chinese Conference on Pattern Recognition and Computer Vision (PRCV). pp. 468–481. Springer (2024)
37. Zhong, Y., Li, B., Tang, L., Kuang, S., Wu, S., Ding, S.: Detecting camouflaged object in frequency domain. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4504–4513 (2022)
38. Zhou, D.W., Zhang, Y., Wang, Y., Ning, J., Ye, H.J., Zhan, D.C., Liu, Z.: Learning without forgetting for vision-language models. IEEE Transactions on Pattern Analysis and Machine Intelligence **47**(6), 4489–4504 (2025)