PGNet: Prototype-Guided Camouflaged object detection

1st Junyong Sun
College of Computer Science and Technology
Zhejiang University of Technology
Hangzhou, China
221125120165@zjut.edu.cn

2nd Yuan Sun
College of Computer Science and Technology
Zhejiang University of Technology
Hangzhou, China
sy02g@zjut.edu.cn

3rd Yongbiao Zhao
School of Science
Zhijiang College of Zhejiang University of Technology
Shaoxing, China
zhaoyb@zjc.zjut.edu.cn

4th Zhihu Zhou
College of Information Engineering
Zhejiang University of Technology
Hangzhou, China
zhouzhihu@zjut.edu.cn

5th Zhuchenghao Wang
College of Computer Science and Technology
Zhejiang University of Technology
Hangzhou, China
211125120035@zjut.edu.cn

6th Keji Mao*
College of Computer Science and Technology
Zhejiang University of Technology
Hangzhou, China
maokeji@zjut.edu.cn

***Abstract*—Camouflaged object detection focuses on detecting objects that blend into complex backgrounds. Compared to general detection tasks, camouflaged targets have stronger similarity to their surrounding background in terms of texture and color, and their boundaries are often blurred, which significantly increases the difficulty of accurate segmentation. To address these issues, we propose a novel network framework: Prototype Guided Network (PGNet). It extracts background information through prototype learning, highlights the foreground, and uses an attention mechanism to make the model focus on the target region of the image. Under the guidance of edge information, it effectively segments the target. Specifically, it consists of three modules: 1) Background Prototype Module (BPM), which can extract a learnable background prototype from the image, suppress camouflaged region features similar to the background, and help distinguish between foreground and background. 2) Prototype Attention Module (PAM), which employs the prototype as a global context and uses overlock attention to make the model pay more attention to the target region. 3) Edge Guided Module (EGM), which predicts the target edges and guides the segmentation of the image, solving the problem of blurred edges. Extensive experiments show that PGNet outperforms the current COD method in terms of segmentation accuracy.**

***Index Terms*—camouflaged object detection, prototype learning, edge guidance**

I. Introduction

To evade predators or improve hunting skills, many organisms have evolved the ability to blend into their environment, becoming difficult to distinguish with the naked eye; this ability is called camouflage. Camouflaged object detection (COD) [1] aims to detect objects that blend into their surroundings from complex backgrounds. It is used in many fields, such as hidden defect detection in industry [2], early pathological tissue discovery in medicine [3].

Targets in Camouflaged object detection are often more challenging than general object detection because they have a high degree of similarity to the background, no clear boundaries, and no obvious contrast with the environment. In recent years, with the development of deep learning, many deep learning-based methods have emerged. These methods automatically learn representations to directly extract complex deep features from data, demonstrating excellent performance in various Camouflaged object detection tasks. Even so, existing methods still have certain limitations. First, Current methods often largely ignore background information. However, the background contains a wealth of information; it can not only determine the location of the detected target but also highlight the target based on the texture and color of the background. Second, the attention distribution in the detected image is uniform. When detecting camouflaged targets, we should focus our attention on the camouflaged object as much as possible [4]. Third, the accuracy of edge detection is insufficient [5], especially in scenes with blurred or camouflaged boundaries.

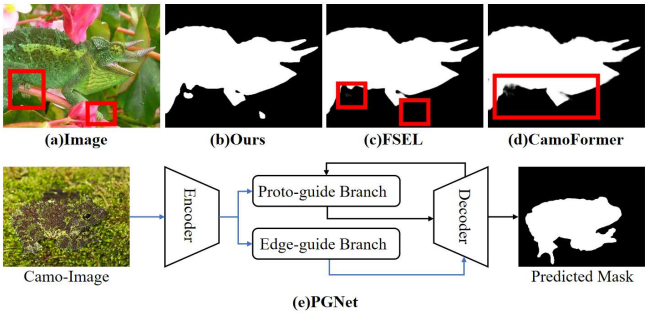

Fig. 1. (a),(b),(c),(d)show a comparison of our PGNet with other methods. (e) is a simplified flowchart of our method, where blue arrows represent coarse predictions and black arrows represent second-round predictions.

Traditional edge detection algorithms often struggle to handle weak or unclear edges.

To address these limitations, we draw inspiration from the observation strategy of biological vision systems, which prioritizes a global perspective over a local one, and propose PGNet. This is a novel multi-round iterative prediction framework. It consists of three modules: a Background Prototype Module (BPM) for highlighting the detected target, a Prototype Attention Module (PAM) to focus the model on the detection region, and an Edge Guided Module (EGM) for generating sharp edges. Specifically, our method first coarsely obtains a coarse mask by predicts the multi-scale features from the encoder. Then, we invert the coarse mask to obtain the background of the image, extract the background prototype information, and perform a second round of prediction under its guidance to highlight the detected target, obtaining a second-round predicted mask. This second-round mask is then combined with the global prototype and generated through an overlock attention [6] mechanism, resulting in dynamic convolutional kernels that assign different weights to different regions, making the model more focused on the detected target region. Simultaneously, we use the EGM to combine multi-scale features with the global semantic context to generate edge predictions [5], addressing the issue of blurred edges. Our contributions are summarized as follows:

- We propose PGNet, a novel multi-round guided prediction framework that performs iterative predictions through multiple rounds guided by background prototypes, attention mechanisms, and edges, enhancing the accuracy and robustness of the prediction model.
- We introduce a prototype-guided approach to image Camouflaged object detection. Previously, prototype-guided learning methods were generally used in few-shot learning domains and were almost non-existent in Camouflaged object detection, providing a new approach for subsequent Camouflaged object detection tasks.
- Extensive experiments on four benchmark datasets using PGNet demonstrate that our model outperforms others in terms of performance, proving the feasibility of our prototype learning, multi-round guided prediction approach.

II. Related Work

A. Camouflaged object detection

Camouflaged object detection is a highly challenging task in computer vision, aiming to identify targets that are highly similar to the environment from complex settings. Traditional Camouflaged object detection methods mainly utilize carefully designed low-level features to capture subtle differences in texture, intensity, and color [7] in images. However, these methods often perform poorly in low-resolution images or when foreground and background have significant visual similarities. Currently, deep learning-based methods have greatly advanced Camouflaged object detection by automatically learning representations to directly extract complex deep features from the data.

Multi-scale aggregation strategy: This strategy enhances the representation by aggregating multi-scale features across different levels, gradually refining the features, enriching their semantic information. For example, CamoFormer [8] is inspired by the Transformer architecture and adopts a mask-separable attention mechanism to refine features at different levels. Zoom-Net [9] uses an attention fusion strategy and leverages scale integration to unify features at different scales and fuse mixed-scale semantics.

Simulating biological strategies: This strategy references the behavior of predators in nature or the human visual detection mechanism. For example, SINet [7] simulates the search and recognition process of predators, first quickly locating areas that may be camouflaged targets in the search phase, and then performing fine-grained differentiation on the output of the search phase in the recognition phase. PreyNet [10] starts from the biological predation mechanism, modeling the detection of camouflaged targets as a dynamic game process between predators and prey to simulate the sensory and cognitive mechanisms of predation.

B. Prototype Learning

Prototype learning involves extracting representative prototypes from the training set and then classifying them by measuring their distance from test samples to supervise future training. This technique is widely used in few-shot learning and some anomaly detection methods also employ prototype learning. However, its application in Camouflaged object detection is currently limited, generally focusing on video camouflage detection. For example, the authors of Camsam2 [11] introduced prototypes into video camouflage segmentation, abstracting mask prototypes from previous frames into prototypes themselves. This provides high-quality temporal priors for subsequent cross-frame temporal feature fusion, ultimately improving the accuracy and temporal consistency of camouflage video segmentation. The authors of R2CNet [4] also utilized the idea of prototype learning. They added a subset of reference images to the training set—salient images representing the same target class. These images were abstracted into common representations of the target class through the reference branch, and these common representations were then used to guide subsequent COD branches. This effectively specifies the category of the reference object for the model, rather than blindly detecting it in images.

In our approach, we utilize the concept of prototype learning in many places. For example, we extract background prototypes in the background prototype module to highlight the camouflaged target; in the prototype attention module, the prototype acts as global semantic information to guide the model to focus on the target.

III. Methodology

In this section, we will present the specific details of the proposed PGNet.

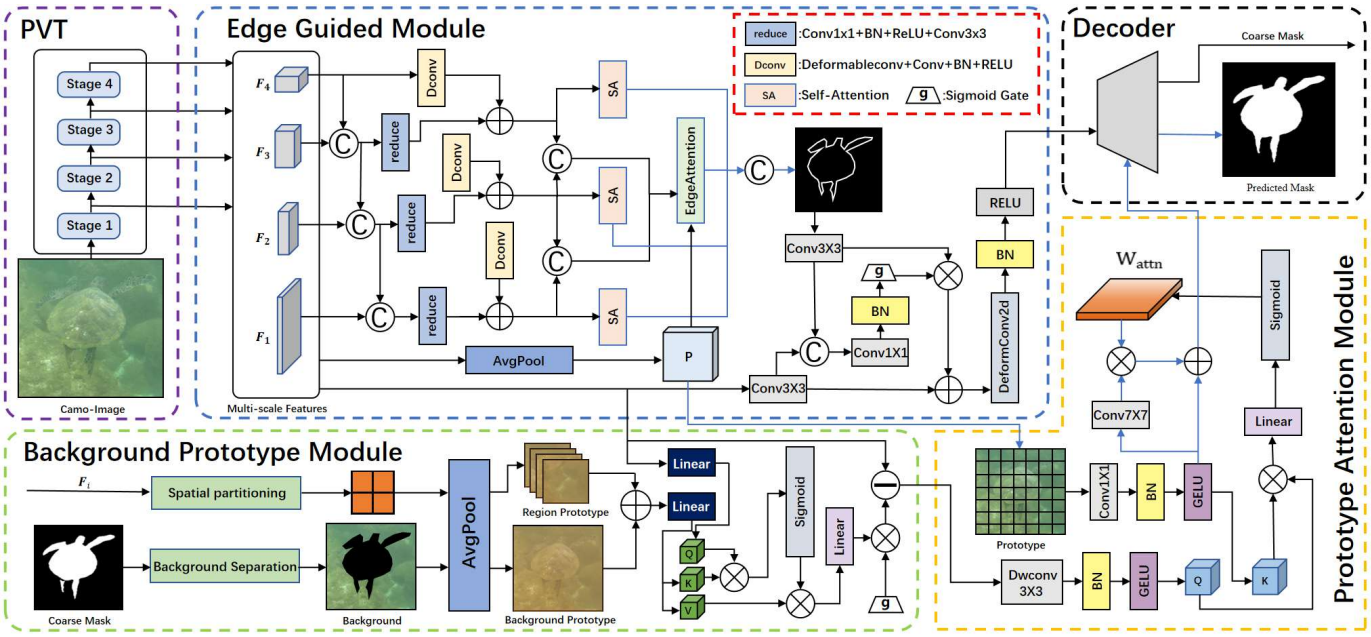

Fig. 2. Our overall framework for PGNet. In the Decoder module, black arrows represent the coarsely predicted input and output, while blue arrows represent the final input and output. Apart from this, the different colored arrows in other modules are only used to distinguish intersections and avoid ambiguity.

A. Overall Architecture

As shown in Fig. 2, similar to most related works, our framework consists of an encoder and multiple components. Each component guides the decoding process, iteratively predicting and refining the mask for progressive decoding.

Encoder: We use PVT-v2 as the encoder. Given an input image I, the encoder effectively extracts multi-scale features (Fi) containing both global and local information.

The other components can be divided into two branches: the prototype-guided branch and the edge-guided branch. The prototype-guided branch consists of a background prototype module and a prototype attention module. The former is used to extract background prototypes and increase foreground-background contrast to highlight the detected target; the latter enables the model to focus on local features at different scales. The edge-guided branch consists of an edge guided module that uses global semantic context to predict the edges of camouflaged targets.

B. Background Prototype Module (BPM)

To better utilize background information and highlight the detection target, the background prototype module first uses multi-scale features Fi for coarse prediction to obtain a coarse mask M_{coarse}, which is then inverted to obtain a background mask M_{bg}. Then, the feature map is spatially partitioned, and masked average pooling is performed on the background and global background images of each region to obtain regional background prototypes and global background prototypes. These are then concatenated to obtain a background prototype that contains both local details and global information such as the texture and color of the background.

$$\mathrm{P}_{\mathrm{bg}} = \frac{\sum_{(h,w)\in \text{region}} F_{h,w} \cdot M_{h,w}}{\sum_{(h,w)\in \text{region}} M_{h,w}} \tag{1}$$

where $F_{h,w}$ represents the feature vector at position (h, w) in the feature map, and $M_{h,w}$ represents the background mask value at (h, w), where the value is 0 or 1, indicating whether (h, w) is a background region.

After obtaining the background prototype, the attention mechanism is applied to enhance the previous feature Fi. Specifically, the process starts by generating Query, Key, and Value, where Q comes from the feature map, K and V come from the background prototype. Then, the similarity between Q and K is calculated to obtain the attention weights α_k. The background representation is obtained by a linear projection (Proj), where each position is determined based on the similarity with the background prototype. Finally, residual gating is applied by subtracting the background representation from the original feature map to obtain the enhanced feature map F_{out}, which is used for the second round of prediction under the guidance of the background prototype.

$$F_{out} = F - \sigma(g) \cdot \mathrm{Proj}\left(\sum_{k=1}^{k} \alpha_k V_k\right) \tag{2}$$

$$\alpha_k = \sigma\left(s \cdot \frac{Q^T \cdot K_k}{\|Q\|\|K_k\|}\right) \tag{3}$$

where σ represents the sigmoid function, k represents the number of background prototypes (four local prototypes and one global prototype), and g represents the learnable gating parameter.

TABLE I
QUANTITATIVE COMPARISON ON THREE COD DATASETS. HIGHER IS BETTER FOR F_β^w, E_ϕ, AND S_m, WHILE LOWER IS BETTER FOR M.

Methods	NC4K (4121)				COD10K (2026)				CAMO (250)			
	$F_\beta^w \uparrow$	$E_\phi \uparrow$	$S_m \uparrow$	$M \downarrow$	$F_\beta^w \uparrow$	$E_\phi \uparrow$	$S_m \uparrow$	$M \downarrow$	$F_\beta^w \uparrow$	$E_\phi \uparrow$	$S_m \uparrow$	$M \downarrow$
SAM2-UNet$_{24}$ [12]	0.856	0.941	**0.901**	0.029	0.789	0.936	0.880	0.021	0.844	0.932	0.884	0.042
CamoFormer$_{24}$ [8]	0.847	0.939	0.892	0.030	0.786	0.932	0.869	0.023	0.831	0.929	0.872	0.046
ACUMEN$_{24}$ [13]	0.826	0.932	0.874	0.036	0.761	0.930	0.852	0.026	0.850	0.939	**0.886**	0.039
VSCode$_{24}$ [14]	0.841	0.935	0.882	0.032	0.780	0.931	0.869	0.025	0.820	0.925	0.873	0.046
FocusDiffuser$_{24}$ [15]	0.854	0.940	0.891	0.029	0.805	0.939	0.875	**0.020**	0.851	0.939	0.881	0.042
FSEL$_{24}$ [16]	0.853	0.941	0.892	0.030	0.800	0.928	0.873	0.021	0.851	0.942	0.885	0.040
SAM-TTT$_{25}$ [17]	**0.893**	**0.946**	0.890	0.033	**0.851**	**0.944**	0.874	0.024	**0.879**	0.938	0.868	0.045
ESCNet$_{25}$ [5]	0.859	0.941	0.892	**0.028**	0.804	0.939	0.873	0.021	0.843	0.934	0.871	0.044
PGNet(Ours)	0.862	**0.946**	0.892	**0.028**	0.807	**0.944**	0.872	**0.020**	0.861	**0.946**	0.883	**0.037**

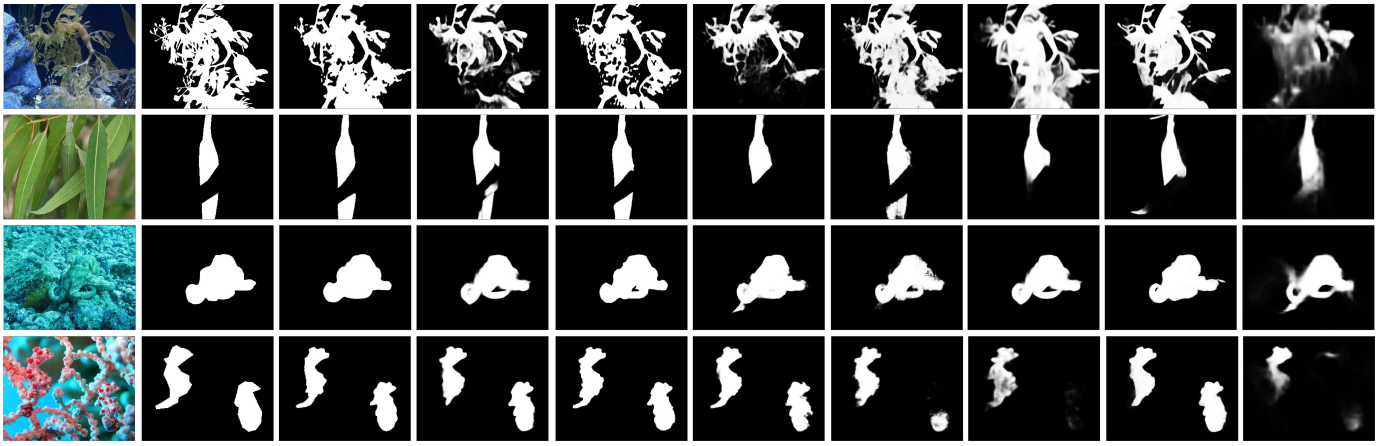

Fig. 3. Visual Comparison of Our PGNet with Current State-of-the-Art Methods.

TABLE II
ABLATION STUDY OF THE PROPOSED MODEL ON THE CAMO DATASETS.

Settings	CAMO (250)			
	$F_\beta^w \uparrow$	$E_\phi \uparrow$	$S_m \uparrow$	$M \downarrow$
Baseline	0.775	0.894	0.832	0.063
EGM	0.813	0.918	0.846	0.051
EGM + PAM	0.854	0.938	0.879	0.039
EGM + PAM + BPM	**0.861**	**0.946**	**0.883**	**0.037**

TABLE III
EXPERIMENTAL RESULTS ON THE CHAMELEON DATASETS.

Settings	CHAMELEON (76)			
	$F_\beta^w \uparrow$	$E_\phi \uparrow$	$S_m \uparrow$	$M \downarrow$
ZoomNet$_{22}$ [9]	0.845	0.952	0.902	0.023
FSPNet$_{23}$ [18]	0.868	0.943	0.908	0.022
FSEL$_{24}$ [16]	**0.880**	**0.958**	**0.916**	0.022
Ours	0.863	0.953	0.897	**0.021**

C. Prototype Attention Module (PAM)

To better enable the model to focus on the detection target, we designed a prototype-guided attention module. The prototype is obtained by directly performing masked average pooling on the feature map. In terms of the attention mechanism, we adopted the idea of overlock [6] attention, first looking at the whole and then the details. We performed convolution, batch regularization and GELU operation on the prototype to obtain F_{global}, which serves as the key to represent the global context. Then, we used dynamic separable convolution on F_{out} to process local detailed features to obtain F_{local}, which serves as the query. We calculated their similarity and generated dynamic spatial attention weights.

$$W_{\text{attn}} = \sigma(\text{Linear}(Q \cdot K^T)) \tag{4}$$

$$F_{last} = F_{local} + W_{\text{attn}} \odot F_{lk} \tag{5}$$

Where $\odot$ denotes element-wise multiplication, $\sigma(\cdot)$ denotes the Sigmoid function.

After obtaining the dynamic spatial attention weights, we expand the receptive field of F_{local} using a large 7×7 convolutional kernel to obtain F_{lk}. We then adaptively fuse local and large kernel features based on the weights, using larger kernel features more extensively in areas with high weights to obtain more contextual information, while preserving more local features in areas with low weights to ensure sufficient detail is retained.

D. Edge Guided Module (EGM)

To address issues such as blurred edges, we designed an edge guided module. In this module, we first predict the target edge. We then fuse the multi-scale feature F_i with the deepest feature F_4 to pass deep semantic information to the shallow layer, enhancing edge detection capabilities. Then, we propagate from top to bottom, using deformable convolution to adaptively adjust the sampling position and better capture the edge shape. Finally, we pass the obtained edge representation f_i through a self-attention module to obtain the deep edge map (e_4), the middle edge map (e_3), and the shallow edge map (e_2).

Following this, we further fused the edge representations pairwise, assigning different edge weights to them through an edge attention mechanism to obtain a detailed edge map (e_1). Finally, we fused the edge information at different scales through upsampling and convolution operations to obtain an edge prediction (E) that has both semantic and detailed characteristics.

$$e_1 = \text{EdgeAttn}([\text{cat}(f_1, f_2);\ \text{cat}(f_2, f_3)]) \tag{6}$$

where $[\cdot;\cdot]$ denotes channel-wise concatenation.

After obtaining the edge prediction, to guide the edge process, we first predict two sets of offsets ($offset_F$ and $offset_E$) from semantic features and edge information, respectively. Since the reliability of edge information varies significantly across different spatial locations, directly fusing the two types of offsets may introduce noise. Therefore, we designed a lightweight gating network to adaptively adjust the impact of edge offsets on the final sampling position. Specifically, the two offsets are concatenated along the channel dimension and input into the gating module:

$$\mathbf{g} = \sigma([\text{offset}_F;\ \text{offset}_E]) \tag{7}$$

$$\text{offset} = \text{offset}_F + \mathbf{g} \odot \text{offset}_E \tag{8}$$

This enables an edge-aware and robust adaptive sampling mechanism. This module effectively improves the model's ability to model complex boundaries and camouflaged targets with almost no additional computational overhead.

E. Loss Function

To simultaneously optimize target region integrity and boundary accuracy during training, we introduce a two-stage, edge-assisted joint loss function to uniformly constrain the model's coarse and fine predictions. The overall loss is defined as:

$$\mathcal{L}_{\text{total}} = 0.5 \times \mathcal{L}_{\text{coarse}} + \mathcal{L}_{\text{refined}} + 0.3 \times \mathcal{L}_{\text{edge}} \tag{9}$$

Where $\mathcal{L}_{\text{coarse}}$ and $\mathcal{L}_{\text{refined}}$ denote the supervision losses for the coarse segmentation prediction and the refined segmentation prediction in the background prototype module, respectively. $\mathcal{L}_{\text{edge}}$ is a multi-scale edge deep supervision term used to guide the model to learn stable and accurate target boundaries. For segmentation supervision, we adopt a structure loss composed of binary cross-entropy (BCE) loss and intersection-over-union (IoU) loss:$\mathcal{L} = \mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{IoU}}$. In addition, Dice loss is employed to supervise edge prediction and multi-scale edge features generated by the edge-guided module are supervised via a deep supervision mechanism:

$$\mathcal{L}_{\text{edge}} = \sum_{i=1}^{4} \mu_i \cdot \mathcal{L}_{\text{dice}}(e_i, GT_{\text{edge}}) \tag{10}$$

Shallow edge predictions are assigned higher weights to enhance fine-grained boundary modeling, ensuring stable training and coordinated coarse-to-fine optimization of segmentation and boundary prediction.

IV. Experiments

A. Experimental Setup

Datasets: Our experiments were evaluated on three benchmark datasets: CAMO [19], COD10K [1], and NC4K [20], which contain a large number of natural and artificially camouflaged scenes. CAMO contains 1250 images, COD10K contains 5066 images, and NC4K contains 4121 images. We used 1000 images from the CAMO dataset and 3040 images from the COD10K dataset for training, and the remaining images were used for testing.

Evaluation Metrics: We use four widely adopted evaluation metrics: Structure-measure, which measures the similarity between the predicted segmentation result and the ground truth label by combining structural and region information, to evaluate image segmentation accuracy; Weighted F-measure, which combines precision and recall to measure the overall performance of classification and localization in object detection; Enhanced-alignment Measure, which is used to comprehensively evaluate the model's performance on all test samples; and Mean Absolute Error, which measures the mean absolute error between the predicted and ground truth values.

Experimental Details: Our model was implemented in PyTorch and trained using two NVIDIA RTX 3080Ti GPUs, with a training time of only one day. The encoder used a PVT-v2-b4 backbone network, with the input image resolution set to 416×416 to match the training settings.

B. Comparison with State-of-the-Arts

Quantitative Evaluation: Table I presents the quantitative results of our proposed method compared with 8 competing models on three benchmark COD datasets. The results demonstrate that PGNet achieves superior performance.

Visual Comparison: Fig. 3 illustrates the comparison between our proposed PGNET and state-of-the-art models. For fair comparison, the results of the compared models were generated from the publicly released code of their authors. The results demonstrate that our method can more accurately predict camouflaged targets, obtaining results with more complete structure, greater detail, and clearer edges.

C. Ablation Studies

Table II presents the ablation study results on the CAMO datasets.

Effectiveness of the Edge Guided Module: As shown in Table II, the results integrated with the edge guided module were comprehensively improved compared to the baseline results. This indicates that the module successfully predicted the target edges and provided guidance for subsequent mask segmentation, preventing edge blurring.

Effectiveness of the Prototype Attention Module: As shown in Table II, integrating the prototype attention module significantly improves model performance, particularly in the MAE. This verifies that the prototype attention module effectively focuses the model on the detected target region.

Effectiveness of the Background Prototype Module: As shown in Table II, introducing the background prototype module further improves model performance. This demonstrates that the module effectively filters out similar attributes in the background, highlighting the camouflaged target.

D. Generalization Analysis

To verify the generalization ability of our model, we tested it on the NC4K and CHAMELEON datasets. As shown in table III, the results showed that our method outperformed other state-of-the-art methods in terms of the mean absolute error (MAE) and maintained high performance on other metrics, thus verifying that our method has excellent generalization ability.

V. Conclusion

In this paper, we propose Prototype-Guided Network (PGNet), introducing the concept of prototype learning to the COD tasks. Through multi-round guidance, iterative prediction generates more refined detection results. We designed a background prototype module to extract background prototypes and combine them with an attention mechanism to highlight the detection target in the second round of prediction. Subsequently, we designed a prototype attention module, using prototypes as global context and enabling the model to focus on sufficient details through overlock attention. In addition, we designed an edge guided module, effectively solving the edge blurring problem by combining multi-scale attention mechanism with deformable convolution. Extensive experimental results show that PGNet performs excellently on benchmark datasets, and ablation experiments also verify the effectiveness of each module. We hope that the concept of prototype learning can promote the development of COD tasks.

Acknowledgments

This research is supported by "Pioneer" and "Leading Goose" R&D Program of Zhejiang Province under No. 2026C02A1019, 2026C02A1222, 2026C02A1018; Zhejiang; Zhejiang Provincial Natural Science Foundation of China under Grant No:LTGG23F020002; Zhejiang University of Technology School-enterprise Cooperation Project under Grant No:KYY-HX-20250798, KYY-HX-20230493, KYY-HX-20250381, KYY-HX-20250423; Medical and Health Science Program of Zhejiang Province, Grant No:WKJ-ZJ-26092.

References

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

[2] A. Kumar, "Computer-vision-based fabric defect detection: A survey," *IEEE transactions on industrial electronics*, vol. 55, no. 1, pp. 348–363, 2008.

[3] H. Chen, P. Wei, G. Guo, and S. Gao, "Sam-cod+: Sam-guided unified framework for weakly-supervised camouflaged object detection," *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.

[4] X. Zhang, B. Yin, Z. Lin, Q. Hou, D.-P. Fan, and M.-M. Cheng, "Referring camouflaged object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[5] S. Ye, X. Chen, Y. Zhang, X. Lin, and L. Cao, "Escnet: Edge-semantic collaborative network for camouflaged object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 20 053–20 063.

[6] M. Lou and Y. Yu, "Overlock: An overview-first-look-closely-next convnet with context-mixing dynamic kernels," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 128–138.

[7] C. Pulla Rao, A. Guruva Reddy, and C. Rama Rao, "Camouflaged object detection for machine vision applications," *International Journal of Speech Technology*, vol. 23, no. 2, pp. 327–335, 2020.

[8] B. Yin, X. Zhang, D.-P. Fan, S. Jiao, M.-M. Cheng, L. Van Gool, and Q. Hou, "Camoformer: Masked separable attention for camouflaged object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[9] 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 *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 2022, pp. 2160–2170.

[10] M. Zhang, S. Xu, Y. Piao, D. Shi, S. Lin, and H. Lu, "Preynet: Preying on camouflaged objects," in *Proceedings of the 30th ACM international conference on multimedia*, 2022, pp. 5323–5332.

[11] Y. Zhou, Y. Li, Y. Fu, L. Benini, E. Konukoglu, and G. Sun, "Camsam2: Segment anything accurately in camouflaged videos," *arXiv preprint arXiv:2503.19730*, 2025.

[12] X. Xiong, Z. Wu, S. Tan, W. Li, F. Tang, Y. Chen, S. Li, J. Ma, and G. Li, "Sam2-unet: Segment anything 2 makes strong encoder for natural and medical image segmentation," *Visual Intelligence*, vol. 4, no. 1, p. 2, 2026.

[13] H. Zhang, Y. Lyu, Q. Yu, H. Liu, H. Ma, D. Yuan, and Y. Yang, "Unlocking attributes' contribution to successful camouflage: A combined textual and visual analysis strategy," in *European Conference on Computer Vision*. Springer, 2024, pp. 315–331.

[14] Z. Luo, N. Liu, W. Zhao, X. Yang, D. Zhang, D.-P. Fan, F. Khan, and J. Han, "Vscode: General visual salient and camouflaged object detection with 2d prompt learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 17 169–17 180.

[15] J. Zhao, X. Li, F. Yang, Q. Zhai, A. Luo, Z. Jiao, and H. Cheng, "Focusdiffuser: Perceiving local disparities for camouflaged object detection," in *European Conference on Computer Vision*. Springer, 2024, pp. 181–198.

[16] Y. Sun, C. Xu, J. Yang, H. Xuan, and L. Luo, "Frequency-spatial entanglement learning for camouflaged object detection," in *European Conference on Computer Vision*. Springer, 2024, pp. 343–360.

[17] Z. Yu, L. Zhao, G. Xiao, and X. Zhang, "Sam-ttt: Segment anything model via reverse parameter configuration and test-time training for camouflaged object detection," in *Proceedings of the 33rd ACM International Conference on Multimedia*, 2025, pp. 4030–4038.

[18] Z. Huang, H. Dai, T.-Z. Xiang, S. Wang, H.-X. Chen, J. Qin, and H. Xiong, "Feature shrinkage pyramid for camouflaged object detection with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 5557–5566.

[19] T.-N. Le, T. V. Nguyen, Z. Nie, M.-T. Tran, and A. Sugimoto, "Anabranch network for camouflaged object segmentation," *Computer vision and image understanding*, vol. 184, pp. 45–56, 2019.

[20] Y. Lv, J. Zhang, Y. Dai, A. Li, B. Liu, N. Barnes, and D.-P. Fan, "Simultaneously localize, segment and rank the camouflaged objects," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 11 591–11 601.