

Highlights

Text-Guided Camouflaged Object Detection

Zefeng Chen, Yunqi Xue, Zhijiang Li, Philip Torr, Jindong Gu

- We introduce an effective method to extract valuable textual information for camouflaged targets from pre-trained multimodal large language models.
- We propose the Text-Guided Camouflaged Object Detection (TG-COD) framework, in which we integrate textual semantic guidance into detection models in both the learning and inference process.
- We validate that our method maintains high performance even in few-shot setting with extremely limited annotated data and demonstrates robust generalization capability.
- Extensive experiments demonstrate the SOTA performance of the TG-COD framework for the challenging COD task, surpassing 18 popular COD approaches across three widely used datasets.

Text-Guided Camouflaged Object Detection

Zefeng Chen^a, Yunqi Xue^a, Zhijiang Li^{a,*}, Philip Torr^b, Jindong Gu^b

^a*School of Information Management, Wuhan University, Wuhan, China*

^b*University of Oxford, Oxford OX1 3PJ, United Kingdom*

Abstract

Camouflaged object detection (COD) aims to identify and segment objects hidden in the background due to their high similarity in color or texture. Recent efforts have investigated the utilization of foundation models to provide extra supervision for solving this task, such as object masks predicted by Segment Anything Models and labeled camouflaged object images generated by Stable Diffusion Models. In this work, instead of visual supervision, we endeavor to utilize Multimodal Large Language Models (MLLM) to provide textual information for COD. Specifically, we propose the Text-Guided Camouflaged Object Detection (TG-COD) framework, which consists of two main stages: extracting pseudo-textual description from MLLM and integrating textual information in the COD process. Our framework leverages the knowledge of MLLM and guides the detection process with textual information. Comprehensive experiments demonstrate that our method achieves state-of-the-art (SOTA) performance and also exhibits strong generalization capability in a few-shot setting.

Keywords: Camouflaged Object Detection, In-context Learning, Multimodal Fusion

1. Introduction

Camouflage is a common defense strategy in the natural world. Organisms can adeptly blend into their surroundings by mimicking the colors, textures, and shapes, thus avoiding detection by predators. Camouflaged object detection (COD) [1, 2] is the

*Corresponding author.

Email addresses: zf.chen038614@outlook.com (Zefeng Chen), yunqi_xue@outlook.com (Yunqi Xue), lizhijiang@whu.edu.cn (Zhijiang Li), philip.torr@eng.ox.ac.uk (Philip Torr), jindong.gu@outlook.com (Jindong Gu)

process of segmenting concealed targets within images, posing a significant challenge in image segmentation tasks. Recently, COD has received increasing interest and has found extensive applications in various domains such as pest detection [3], medical image segmentation [4], and industrial defect detection [5].

Prior studies have focused on designing diverse network architectures to effectively leverage the subtle cues of camouflaged objects [1, 6]. However, the demanding nature of high-quality annotation of camouflaged objects results in a scarcity of annotated data. The performance of these methods is constrained by the limitation of available annotated data. Recently, large foundation models [7] pretrained on extensive datasets have been explored to provide additional supervision for COD tasks. For example, the Segment Anything Model [8] can segment images to provide pseudo-labels [9], thus promoting weakly-supervised COD; Diffusion model [10] can generate labeled data, thereby partially overcoming the challenge of insufficient annotated data.

In this work, we explore the textual supervision offered by multimodal large language models (MLLM) and integrate textual information into the learning and detection process. Specifically, we propose the Text-Guided Camouflaged Object Detection (TG-COD) framework, which enhances the model’s comprehension of camouflaged images with auxiliary textual guidance and leads to more satisfactory results.

Our TG-COD framework comprises two stages. The first stage is to extract valuable semantic information from MLLM. However, simply querying MLLM with a camouflaged object image does not lead to a precise description of the object. A potential reason is that MLLM also struggle with recognizing the camouflaged object in standard references since their pretraining datasets contain almost negligible camouflaged images. To overcome the challenge, we introduce an effective in-context learning-based method to extract useful textual information. Concretely, as depicted in Fig. 1, we provide demonstration images and template text to facilitate inferencing, enabling the MLLMs to accurately comprehend task objectives and generate precise textual descriptions from complex camouflaged images.

In the second stage, we aim to integrate our pseudo-textual description extracted by MLLM into the camouflage detection model. We employ pretrained Segment Anything Model (SAM) as detection model since it can understand text input. To adapt the

pretrained model to COD task, we fine-tune the model using adapters [11] and propose an effective way to integrate textual information into the visual backbone. Similarly, during the inference stage, we extract textual information with MLLM and utilize the generated text to guide the detection process.

We conduct comprehensive experiments, exploring the effects of various text extraction methods and model fine-tuning architectures, showcasing the efficacy of the TG-COD framework. Our approach surpasses existing models, achieving state-of-the-art performance on three widely used datasets. Additionally, our TG-COD demonstrates robust performance in a few-shot setting, successfully generalizing from base classes to novel classes.

Our contributions can be summarized as follows:

- We introduce an effective method to extract valuable textual information for camouflaged targets from pre-trained multimodal large language models.
- We propose the Text-Guided Camouflaged Object Detection (TG-COD) framework, in which we integrate textual semantic guidance into detection models in both the learning and inference process.
- We validate that our method maintains high performance even in few-shot setting with extremely limited annotated data and demonstrates robust generalization capability.
- Extensive experiments demonstrate the SOTA performance of the TG-COD framework for the challenging COD task, surpassing 18 popular COD approaches across three widely used datasets.

2. Related Work

2.1. Camouflaged Object Detection

Deep learning-based approaches for COD have been extensively studied. These approaches focus on designing complex architectures and researching data mining strategies to enhance the performance of COD. These generally include the following as-

pects: multi-scale feature fusion [6, 12], multi-stage refinement [13], low-level information fusion [14, 15], foreground-background separation [16, 17], uncertainty [18], graph learning [19], and attention mechanisms [20]. However, these specially designed model architectures and data mining strategies often have limited improvements on limited annotated data.

2.2. Foundation Models

Foundation Models have been explored for COD in two ways: 1) providing pseudo supervision or 2) solving COD task directly. Large foundation models pretrained on extensive datasets can provide additional supervision for COD tasks. Recent work [9] has attempted to utilize SAM to generate pseudo-labels for camouflaged images. SCODE [21] proposed a camouflage object image synthesis framework based on diffusion models to synthesize camouflaged images, thereby expanding the camouflage image dataset. CamDiff [10] achieves camouflage image enhancement by synthesizing salient objects in camouflage scenes using a latent diffusion model, thus generating camouflage samples with richer features. Different from the aforementioned work that utilizes foundation models to generate visual data, we explore the feasibility of obtaining pseudo-text supervision from MLLM.

The high-level semantic information extraction capability of the diffusion model [22] can be used to predict masks directly for COD. DiffCOD [23] and CamoDiffusion [24] treat COD as a special conditional generation task. Additionally, as model sizes continue to increase, multimodal large language models (MLLM) [25, 26] are capable of generating reasonable descriptions based on images and textual instructions. Efforts [27] have been made to utilize MLLM to predict the position of camouflage objects. However, these methods struggle to achieve accurate pixel-wise predictions, resulting in limited performance improvements. A more relevant work is SAM-adapter [28], which utilizes adapters to integrate domain-specific information or visual prompts into the segmentation network. Differing from theirs, our work focuses on creating pseudo-text guidance and integrating them into SAM.

Algorithm 1 Pseudo-textual Description Extraction.

Input: query dataset Q , support set S , vision encoder E_v , similarity calculation Sim , number of demos N

Output: generated text T

```
1: Initialization:  $T \leftarrow []$  ;
2: for query  $q \in Q$  do
3:   Initialization:  $C \leftarrow []$  ;
4:    $s \leftarrow Sim(E_v(q), E_v(S))$  ;
5:    $C \leftarrow$  choose Top- $N$  demos from  $S$  based on  $s$  ;
6:    $text \leftarrow$  multiple inference and select the text based on transition score ;
7:    $T += text$  ;
8: end for
```

second stage, we tokenize the generated texts using a pre-trained language encoder and feed them into the vision encoder of SAM. During training, the encoder is kept frozen and equipped with learnable adapters. The fine-tuning with the adapters aims to integrate textual features without compromising the performance by modifying its representation space. More about the integration and adapter finetuning are in Sec. 4. We now present the details of our two stages.

3.1. Pseudo-Textual Description Extraction

An intuitive way to extract pseudo-textual description is to query MLLM with a camouflaged object image and a question directly, e.g., "*Please describe the concealed object in the image.*". However, the solution does not work well since MLLMs have also difficulty recognizing concealed objects in standard inference.

To extract useful textual information, we introduce an effective in-context learning (ICL)-based method. ICL[31] enables the models to quickly adapt to a specific context given a set of demonstrations, without any model parameter updates. The demonstrations could be (image, description) pairs in multimodal settings. Conditioning on the demonstrations, MLLMs with ICL ability can output desired information following the format of demonstrations. Specifically, as shown in the first stage (left) of Fig. 1, image and text pairs are selected from a support dataset as demos, and OpenFlamingo outputs

textual description for the camouflaged image conditioning on the selected demos. We describe the creation of the support dataset, demonstration selection, as well as our ICL-based inference pipeline as follows.

Support Dataset Creation. The support dataset is built on the images of the COD10K training set. For each image, we generate a textual description based on its ground-truth mask. Concretely, the size, category, and location details are extracted from the ground truth and represented in natural language. The size of the camouflaged object is determined by the proportion of foreground pixels to the entire image, its location is determined based on the object’s position, and the annotated dataset’s category serves as the category information, formulated into captions using a predefined template (e.g., *The medium size butterfly is camouflaged in the center of the image.*). We opt for these three types of semantic knowledge due to their direct availability from the dataset annotations or their low-cost acquisition through masks. Incorporating richer semantic knowledge could further enhance the model’s comprehension of camouflaged object images.

Demonstration Selection. Given a support dataset S and a query image, candidates are selected based on image similarity, as shown in Alg. 1. In our approach, the training set of the COD10K dataset, annotated with category labels, serves as the support set S . Both the images from the training and test sets serve as the query set Q . For each q in the query set, N context demonstrations $\{s_1, s_2, \dots, s_n\}$ are chosen from the support set. Initially, captions are obtained from the ground truth. These demos benefit context learning in two ways. First, the samples provide a similar data distribution, aiding in the estimation of query labels. Second, the annotated text offers cues for the COD task.

ICL-based Inference Pipeline. ICL-based inferences described above do not always generate meaningful outputs. To further enhance the quality of generated texts, a sampling-based generation approach is employed. Specifically, multiple inferences are conducted for each query image, and the one with the highest score (i.e. perplexity score[32]) is selected. In each inference, the query camouflage image is cropped differently. Fixed cropping at the top-left, top-right, bottom-left, bottom-right, and center positions of the image is utilized. The final description is also adapted with the corresponding cropping.

Overall, our ICL-based inference pipeline can be summarized in Alg. 1. CLIP image encoder is utilized to extract the visual features q_{img} of the query image and S_{img} of the support dataset. Based on visual feature similarity $Sim(q_{img}, S_{img})$, the N most similar samples are selected as demos from the support dataset.

Why not use textual information from ground-truth mask for training? Although it's possible to directly extract text from the ground truth mask annotation, it's not feasible to obtain annotations during the inference stage. Therefore, we employ MLLM to generate text for training data so that the trained model generalizes well to test data with pseudo textual description provided by the same MLLM.

3.2. Text-Guided Camouflaged Object Detection

In the last section, we have described how to extract the textual description of camouflaged object images with MLLMs. We now present how to integrate the extracted textual information into the COD process. We adopt Segment Anything Model (SAM) for COD since it can understand textual input prompts, which makes it a suitable model for the integration of our textual information. Although exhibiting impressive segmentation performance, pre-trained SAM appears less proficient in hidden scenes such as camouflage animals [33]. To adapt it to COD task, we propose to integrate textual information into SAM vision encoder and fine-tune SAM with learnable adapters.

For the integration of textual information, CLIP text encoder is applied to extract the embedding of the provided text, which encodes generated text into a single token. We opt to avoid designing complex network structures that could impose computational burden. Hence, we utilize simple multiplication operations to inject textual semantic knowledge. The architecture is illustrated in the Fig. 2 (a). The embedding after textual integration can be expressed as follows:

$$x'_l = MLP_s(MLP_l(t_l)) \cdot x_l + x_l \quad (1)$$

where MLP_s is a shared MLP layers across all layers, each consisting of two learnable fully connected layers with a GELU activation in between. MLP_l is the l -th learnable MLP layer, $t_l = MLP_l(t_{l-1})$ is the text token output of the l -th MLP layer, and x_l is the

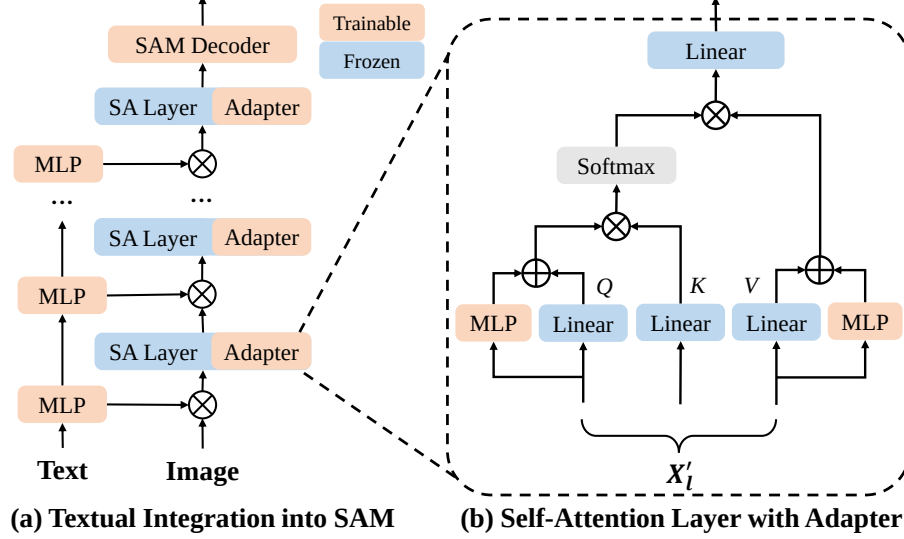


Fig. 2: (a) A series of parallel MLP layers are used to inject semantic information into the image features of each layer of SAM. (b) Parallel MLP layers are connected to adjust the self-attention pattern.

output of the l -th encoder block before textual integration. The integration is conducted before each self-attention block.

We first recall self-attention layers in the SAM vision encoder. Each self-attention block mainly consists of two types of sub-layers: multi-head self-attention (MHSA) layers and MLP layers. In MHSA, tokens are linearly projected (i.e. $W(\cdot)$) and further reformulated into three vectors: $Q = W_q(x_l)$, $K = W_k(x_l)$, and $V = W_v(x_l)$. The MHSA can be expressed as follows:

$$\bar{x}_{l+1} = \text{Attention}(Q, K, V) = \text{Softmax}(QK^T / \sqrt{d})V \quad (2)$$

where d is the dimensionality of the vectors, and $\text{Softmax}()$ is the normalization activation function. T is the transpose operation. We now show our self-attention layer of SAM with adapters and our new mixture features in Fig. 2. The subfigure b shows the details of each adapted self-attention layer. Specifically, we adjust the model's self-attention pattern for camouflage tasks by appending an additional learnable adapter (i.e., an MLP layer) to both Q and V individually. This process is formalized as fol-

lows:

$$Q' = MLP_q(x_l') + W_q(x_l'), \quad V' = MLP_v(x_l') + W_v(x_l') \quad (3)$$

$$\tilde{x}_{l+1} = Attention(Q', K, V') + x_l', \quad (4)$$

where x_l' represents the output of Eq. 1, i.e. mixed features, W_q and W_v , are kept frozen, MLP_q and MLP_v , are additional learnable module. After the self-attention, standard MLP and Layer Normalization are applied, the final output of the self-attention block is $x_{l+1} = MLP(LN(\tilde{x}_{l+1})) + \tilde{x}_{l+1}$.

During training, the pseudo-textual descriptions of training images are prepared in the first stage. COD in the second stage adopts an end-to-end training approach, using binary cross-entropy loss and IOU loss as the loss functions. Meanwhile, textual information is integrated as auxiliary features. During inference, we first extract a description for a given query image with the same MLLM and conduct inference with the fine-tuned SAM model under the guidance of extracted textual information.

4. Experiments

Datasets. We evaluate our methods on three popular COD benchmark, including CAMO [34], COD10K [1], and NC4K [35]. The training and test set settings follow the SINet [1]. In the few-shot setting, we select 42 categories from the COD10K dataset as base classes, leaving the remaining 27 categories as novel classes for testing zero-shot generalization capability. Each base class contains 10 images, resulting in a total of 420 training images. The test set is divided into 1,473 images belonging to the base class (COD10K-Base) and 553 images belonging to the novel class (COD10K-Novel).

Evaluation Metrics. We adopt four evaluation metrics for evaluating predicted masks, including S-measure (S_α) [36], E-measure (E_ϕ) [37], weighted F-measure (F_β^ω) [38], and mean absolute error (MAE) [39].

Implementation Details. In the stage of extracting pseudo textual description, we utilize OpenFlamingo [31] to generate text and leverage CLIP encoder [40] to embed image and text into a unified image-text space for similarity calculation. In the stage of text-guided camouflaged object detection, we employ CLIP’s text encoder to encode the textual descriptions into text embeddings, offering semantic guidance. We utilize

Tab. 1: Quantitative comparison with 18 SOTA methods on three benchmark datasets. Ours outperforms all. Notes $\uparrow$ ($\downarrow$) denote the larger (smaller) is better. – means the result is not available. The best and second best are **bolded** and underlined for highlighting, respectively.

Methods	CAMO (250)				COD10K (2,026)				NC4K (4,121)			
	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$
SINet [1]	0.746	0.774	0.632	0.100	0.774	0.818	0.553	0.049	0.811	0.849	0.689	0.063
PFNet [16]	0.772	0.830	0.697	0.087	0.797	0.875	0.647	0.040	0.828	0.886	0.753	0.053
SINet-V2 [2]	0.808	0.862	0.737	0.075	0.816	0.889	0.669	0.036	0.847	0.901	0.774	0.047
FAPNet [14]	0.799	0.851	0.714	0.080	0.821	0.886	0.682	0.035	0.849	0.897	0.772	0.048
BSA-Net [41]	0.806	0.860	0.738	0.077	0.816	0.887	0.699	0.034	0.841	0.892	0.782	0.048
BGNet [42]	0.811	0.868	0.745	0.072	0.827	0.895	0.694	0.033	0.849	0.902	0.778	0.046
ZoomNet [6]	0.809	0.857	0.738	0.070	0.837	0.894	0.728	0.028	0.856	0.910	0.790	0.042
MFFN [43]	0.811	0.859	0.731	0.075	0.826	0.870	0.710	0.035	0.837	0.875	0.767	0.051
HitNet [44]	0.835	0.889	0.787	0.060	0.857	<u>0.927</u>	0.788	<u>0.025</u>	0.864	0.915	0.818	0.040
FSPNet [45]	<u>0.857</u>	0.901	0.803	0.051	0.849	0.895	0.733	0.027	0.878	0.916	0.820	<u>0.035</u>
EVP [46]	0.847	0.897	0.781	0.058	0.842	0.906	0.742	0.029	0.872	0.917	0.810	0.039
PopNet [47]	0.806	0.857	0.748	0.077	0.849	0.910	0.756	0.028	0.863	0.912	0.805	0.041
PENet [48]	0.826	0.885	0.770	0.064	0.828	0.901	0.719	0.032	0.853	0.908	0.794	0.043
SAM-Adapter [28]	0.851	0.901	0.781	0.068	<u>0.884</u>	0.917	0.805	<u>0.025</u>	<u>0.887</u>	0.912	0.821	0.040
SPANet [49]	0.822	0.883	0.772	0.070	0.835	0.905	0.727	0.031	0.849	0.906	0.788	0.045
BTTNet [50]	0.807	0.861	0.778	0.077	0.816	0.888	0.732	0.034	-	-	-	-
BiCOD [51]	-	-	-	-	0.820	0.879	<u>0.827</u>	0.032	0.827	0.865	0.793	0.055
HAITNet [52]	0.855	0.922	<u>0.823</u>	<u>0.050</u>	0.846	0.924	0.773	<u>0.025</u>	0.871	<u>0.926</u>	<u>0.840</u>	0.034
TG-COD(Ours)	0.876	<u>0.916</u>	0.830	0.048	0.895	0.940	0.834	0.021	0.892	0.927	0.849	0.034

ViT-B version of SAM as the vision model. Other modules are initialized randomly. We follow the training set settings in SINet [1], and apply random flipping, random cropping, and color augmentation to augment the training data. The input images is resized to SAM’s default resolution 1024×1024 [8]. The loss function comprising BCE loss and IOU loss is applied. The AdamW optimizer is applied to all experiments, with an initial learning rate of 2×10^{-4} and cosine learning rate decay strategy. The total training epoch are set to 60, with a batch size of 2. All our experiments are conducted on an NVIDIA GeForce RTX 3090 GPU.

4.1. Comparison with State-of-the-Art COD Methods

We compare our TG-COD with 18 state-of-the-art methods including SINet [1], PFNet [16], SINet-V2 [2], FAPNet [14], BSA-Net [41], BGNet [42], ZoomNet [6],

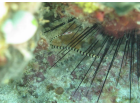
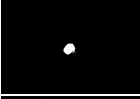
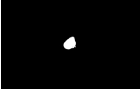
Text	The small size spider is camouflaged in the center of the image.	The small size pipefish is camouflaged in the top of the image.	The medium size ghostpipefish is camouflaged in the center of the image.	The medium size katydid is camouflaged in the top-right of the image.	The small size dog is camouflaged in the center of the image.	The small size toad is camouflaged in the center of the image.
Image						
GT						
Adapter						
Ours						

Fig. 3: Visualization of ablation experiment. "Text" is the generated guiding text. "Adapter" means training the model without the guidance of the generated text. Our TG-COD with text guidance can generate more precise and complete predictions.

MFFN [43], HitNet [44], FSPNet [45], EVP [46], PopNet [47], PENet [48], SAM-Adapter [28], SPANet [49], BTTNet [50], BiCOD [51] and HAITNet [52]. Following their configurations, the results of all these methods are reproduced with their corresponding source code on our comprehensive evaluation datasets.

Quantitative Comparison. Tab. 1 summarizes the quantitative results of our proposed method against 18 competitors on three COD benchmark datasets under four evaluation metrics. Our method surpasses all other models in three popular datasets.

Visualization. We visualize the results of SAM-Adapter (training the model only with the adapter) and our TG-COD (training with text guidance) in Fig. 3. More visualization results are presented in Tab. 2. Textual semantics provide the model with supplementary domain knowledge, enabling the model to perceive the target more accurately. By combining implicit semantic information, such as inherent morphological characteristics of specific animals and details regarding the scale and position of targets within the image, the model avoids focusing on confusing background objects and enhances its ability to identify disguised targets. For instance, in the first column, our model adeptly refrains from segmenting perplexing tree leaves. Additionally, textual semantics aid the

Tab. 2: Under the few-shot setting, our method exhibits superior performance and generalization compared to five competitive COD methods. * means training under the few-shot setting. "-Base" ("-Novel") denotes images of classes include (not include) in the few-shot training dataset. (42, 420, 1473) represent the number of training categories, the number of training samples, and the number of testing samples in the base dataset, respectively. (27, 553) represent the number of testing categories and the number of testing samples in the novel dataset, respectively.

Methods	COD10K-Base (42, 420, 1473)				COD10K-Novel (27, 553)			
	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$
BSANet*	0.750	0.825	0.585	0.050	0.709	0.784	0.521	0.057
FAPNet*	0.758	0.833	0.599	0.049	0.728	0.801	0.533	0.054
EVP*	0.802	0.869	0.662	0.040	0.772	0.836	0.603	0.048
FSPNet*	<u>0.811</u>	0.868	0.647	0.037	<u>0.789</u>	0.845	0.606	0.043
HiNet*	0.806	<u>0.888</u>	<u>0.706</u>	<u>0.036</u>	0.784	<u>0.861</u>	<u>0.664</u>	<u>0.040</u>
Ours*	0.855	0.903	0.760	0.031	0.835	0.880	0.722	0.035

model in producing more exhaustive segmentations, as seen in the third column where the identification of ghost pipefish in the image is more comprehensive. Size and position cues aid in locating and discovering small-scale targets, as demonstrated in the fifth and sixth columns where the model accurately identifies tiny targets. Descriptive natural language accounts of camouflaged targets equip the model with additional rich knowledge concerning localization and recognition, potentially including implicit information about target category morphology, color, and appearance. This contributes to the segmentation model’s perception of target integrity and precision.

Generalization Capability. Auxiliary text guidance enriches the model with additional semantic knowledge, thereby enhancing the SAM model’s comprehension of concealed objects. Namely, the text guidance augments the model’s generalization ability. Two types of generalization ability are considered: 1) the model’s performance when trained on very limited training data, and 2) the model’s performance on novel classes that are not seen in the training. To evaluate this, we train the model on a selected few-shot dataset and evaluate its performance on both the base classes (concealed objects seen in training) and the novel classes (concealed objects unseen in training). For comparison, we also select five powerful few-shot COD models as competitors. Our TG-COD emphasizes model’s performance and generalization ability with extremely limited train-

Tab. 3: Under the few-shot setting, our method exhibits superior performance and generalization compared to five competitive COD methods. * means training under the few-shot setting. "-Base" ("-Novel") denotes images of classes include (not include) in the few-shot training dataset. (36, 360, 1276) represent the number of training categories, the number of training samples, and the number of testing samples in the base dataset, respectively. (33, 750) represent the number of testing categories and the number of testing samples in the novel dataset, respectively.

Methods	COD10K-Base(36,360,1276)				COD10K-Novel(33,750)			
	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$
BSANet*	0.742	0.820	0.571	0.048	0.724	0.794	0.549	0.061
FAPNet*	0.762	0.839	0.579	0.048	0.743	0.810	0.557	0.060
EVP*	0.795	0.859	0.649	0.039	0.786	0.842	0.632	0.047
FSPNet*	<u>0.804</u>	0.866	0.621	0.037	<u>0.794</u>	0.848	0.615	<u>0.046</u>
HitNet*	<u>0.804</u>	<u>0.885</u>	<u>0.689</u>	<u>0.035</u>	0.777	<u>0.851</u>	<u>0.642</u>	0.049
Ours*	0.852	0.901	0.754	0.029	0.842	0.882	0.734	0.036

ing data. As shown in Tab. 2, even when training images are reduced to nearly 1/10 of their original size (from 4040 to 420 instances) on the dataset, our method maintains a high level of performance and significantly outperforms other competitors. Specifically, in base class, indicators S_α , E_ϕ , F_β^ω , and MAE gain increments of 5.4%, 1.7%, 7.6%, and 13.9%, respectively. In novel class, performant gains of 5.8%, 2.2%, 8.7%, and 12.5% are observed in these indicators, which verify the high generalization ability of our TG-COD.

We conducted additional experiments with various categories and varying numbers of shots to further demonstrate the robust generalization capability of our proposed TG-COD. Experimental results shown in Tabs. 3 to 6 demonstrate that under different experimental settings, our TG-COD method consistently maintains advantages. These results indicate that our method consistently outperforms competitors in terms of generalization, rather than being attributed to specific numbers of categories or training images. The discussion above illustrates the effectiveness of the learning strategy that preserves the general capabilities of the foundation model and integrates domain-specific knowledge for downstream tasks.

Tab. 4: Under the few-shot setting, our method exhibits superior performance and generalization compared to five competitive COD methods. * means training under the few-shot setting. "-Base" ("-Novel") denotes images of classes include (not include) in the few-shot training dataset. (30, 300, 1026) represent the number of training categories, the number of training samples, and the number of testing samples in the base dataset, respectively. (39, 1000) represent the number of testing categories and the number of testing samples in the novel dataset, respectively.

Methods	COD10K-Base(30,300,1026)				COD10K-Novel(39,1000)			
	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$
BSANet*	0.753	0.831	0.588	0.044	0.712	0.786	0.540	0.062
FAPNet*	0.764	0.842	0.582	0.045	0.725	0.799	0.538	0.063
EVP*	0.801	0.867	0.654	0.038	0.775	0.834	0.611	0.050
FSPNet*	<u>0.806</u>	0.862	0.606	0.036	<u>0.784</u>	<u>0.841</u>	0.592	<u>0.048</u>
HitNet*	0.800	<u>0.872</u>	<u>0.692</u>	<u>0.034</u>	0.764	0.836	<u>0.642</u>	<u>0.048</u>
Ours*	0.857	0.904	0.754	0.027	0.824	0.866	0.709	0.040

Tab. 5: Under the few-shot setting, our method exhibits superior performance and generalization compared to five competitive COD methods. * means training under the few-shot setting. "-Base" ("-Novel") denotes images of classes include (not include) in the few-shot training dataset. (36, 288, 1276) represent the number of training categories, the number of training samples, and the number of testing samples in the base dataset, respectively. (33, 750) represent the number of testing categories and the number of testing samples in the novel dataset, respectively.

Methods	COD10K-Base(36,288,1276)				COD10K-Novel(33,750)			
	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$
BSANet*	0.735	0.815	0.562	0.048	0.717	0.795	0.543	0.060
FAPNet*	0.757	0.837	0.591	0.047	0.733	0.800	0.552	0.060
EVP*	0.790	0.853	0.634	0.040	0.785	0.841	0.623	0.049
FSPNet*	<u>0.794</u>	0.854	0.609	0.039	<u>0.786</u>	0.835	0.608	0.048
HitNet*	0.791	<u>0.874</u>	<u>0.677</u>	<u>0.037</u>	0.768	<u>0.847</u>	<u>0.643</u>	<u>0.047</u>
Ours*	0.845	0.893	0.737	0.031	0.831	0.874	0.711	0.039

4.2. Ablation Studies and Analysis

In this section, we perform comprehensive ablation experiments and analysis to validate the efficacy of our TG-COD.

Effectiveness of Model Components. We conduct ablation experiments on different components of the proposed model to validate their effectiveness. Specifically, we se-

Tab. 6: Under the few-shot setting, our method exhibits superior performance and generalization compared to five competitive COD methods. * means training under the few-shot setting. "-Base" ("-Novel") denotes images of classes include (not include) in the few-shot training dataset. (36, 144, 1276) represent the number of training categories, the number of training samples, and the number of testing samples in the base dataset, respectively. (33, 750) represent the number of testing categories and the number of testing samples in the novel dataset, respectively.

Methods	COD10K-Base(36,144,1276)				COD10K-Novel(33,750)			
	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$
BSANet*	0.696	0.747	0.497	0.059	0.684	0.757	0.485	0.071
FAPNet*	0.735	0.803	0.539	0.052	0.723	0.789	0.529	0.062
EVP*	<u>0.763</u>	0.815	0.584	0.049	0.764	0.818	<u>0.577</u>	0.057
FSPNet*	0.773	<u>0.831</u>	0.541	0.048	<u>0.774</u>	<u>0.829</u>	0.553	<u>0.055</u>
HitNet*	0.754	0.827	<u>0.618</u>	<u>0.044</u>	0.717	0.784	0.559	0.059
Ours*	0.822	0.870	0.691	0.038	0.805	0.838	0.666	0.047

Tab. 7: Ablation experiments on the components of our Network. "FT" means full fine-tune. "DE," "AD," and "TG" means freezing the backbone parameters and training the parameters of the "Decoder," "Adapter," and "Textual Guide," respectively.

Methods	CAMO (250)				COD10K (2,026)				NC4K (4,121)			
	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$
FT	0.601	0.614	0.327	0.195	0.571	0.588	0.202	0.157	0.611	0.621	0.313	0.174
DE	0.774	0.786	0.634	0.106	0.800	0.842	0.634	0.048	0.814	0.840	0.686	0.071
DE+TG	0.856	0.892	0.785	0.062	0.870	0.914	0.781	0.027	0.878	0.912	0.814	0.040
DE+AD	<u>0.861</u>	<u>0.905</u>	<u>0.812</u>	<u>0.055</u>	<u>0.883</u>	<u>0.928</u>	<u>0.817</u>	<u>0.023</u>	<u>0.882</u>	<u>0.916</u>	<u>0.835</u>	<u>0.037</u>
DE+TG+AD	0.876	0.916	0.830	0.048	0.895	0.940	0.834	0.021	0.892	0.927	0.849	0.034

quentially equip our Text Guidance and Adapter modules into the baseline model, where only the decoder parameters of SAM are trainable. The experimental results in Tab. 7 demonstrate that both Text Guidance and Adapter contribute to the model's performance.

Ablation on Fusion Methods. We explore different approaches to integrate textual features, primarily focusing on linear mappings (Add, Mul, Add+Mul), along with common cross-attention mechanisms. The results presented in Tab. 8 suggest that the performance of diverse fusion methods is comparable. This observation may arise from

Tab. 8: Ablation experiments on different text integration methods. "Add," "Mul," "Cross Attention," and "Add+Mul" represent different approaches to integrating text information into the visual model.

Methods	CAMO (250)				COD10K (2,026)				NC4K (4,121)			
	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$
Add	0.869	0.909	0.813	0.052	0.894	0.936	0.831	0.021	0.889	0.923	0.846	0.036
Mul	0.876	0.916	0.830	0.048	0.895	0.940	0.834	0.021	0.892	0.927	0.849	0.034
Cross Attention	0.874	0.914	0.821	0.048	0.892	0.937	0.829	0.021	0.890	0.924	0.847	0.035
Add+Mul	0.873	0.914	0.827	0.049	0.896	0.939	0.831	0.020	0.893	0.927	0.847	0.034

the adapter’s inherent ability to effectively adjust the model’s representation space for downstream tasks. At the same time, SAM can adeptly utilize text prompts, aligning the semantic space of text within the model itself. Consequently, simple linear mappings can produce satisfactory results without requiring higher-order mapping relationships. Given the comparable performance across different methods of integrating textual semantics, we prioritize a balance between model simplicity and performance, ultimately selecting the Mul configuration.

Ablation on Different Layers. Ablation experiments are conducted to assess the effects of modifying the model across different layers of the SAM. Specifically, we add our structure to the following layers: local layers $\{L_2, L_5, L_8, L_{11}\}$, global layers $\{L_3, L_6, L_9, L_{12}\}$, shallow layers $\{L_1, L_2, \dots, L_6\}$, deep layers $\{L_7, L_8, \dots, L_{12}\}$, and all layers $\{L_1, L_2, \dots, L_{12}\}$. The results shown in Tab. 9 indicate that optimal performance is achieved through the adjustment of features across all layers. The performance of shallow and deep layers indicates that adapting the deeper layers of the neural network leads to superior performance. Additionally, modifying only the global layers can achieve satisfactory performance. The results indicate the importance of adapting high-level semantics features and capturing long-range context for the COD task.

Ablation on Different resolution. Given that SAM’s default input resolution is 1024x1024, which exceeds that of most COD models, we examined the performance of the proposed method at various input sizes. Experimental results in Tab. 10 show that our approach surpasses competitors at similar resolutions. Additionally, increasing the resolution improves performance. This enhancement is partly due to higher resolutions

Tab. 9: Ablation experiments on different layers of the trained model. L_i denotes training the i -th layer.

Methods	CAMO (250)				COD10K (2,026)				NC4K (4,121)			
	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$
$\{L_2, L_5, L_8, L_{11}\}$	0.865	0.906	0.807	0.053	0.887	0.933	0.818	0.022	0.885	0.920	0.835	0.037
$\{L_3, L_6, L_9, L_{12}\}$	0.868	0.907	0.808	0.053	0.892	0.937	0.826	0.022	0.887	0.921	0.841	0.036
$\{L_1, L_2, \dots, L_6\}$	0.845	0.887	0.775	0.063	0.868	0.916	0.782	0.027	0.869	0.904	0.805	0.043
$\{L_7, L_8, \dots, L_{12}\}$	0.866	0.908	0.803	0.056	0.890	0.936	0.827	0.021	0.885	0.921	0.837	0.037
$\{L_1, L_2, \dots, L_{12}\}$	0.876	0.916	0.830	0.048	0.895	0.940	0.834	0.021	0.892	0.927	0.849	0.034

Tab. 10: Ablations on various input resolution. The models that perform best at the same resolution are selected as competitors.

Methods	Resolution	CAMO (250)				COD10K (2,026)				NC4K (4,121)			
		$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$
FSPNet	384*384	0.857	0.901	0.803	0.051	0.849	0.895	0.733	0.027	0.878	0.916	0.820	0.035
Ours	384*384	0.872	0.915	0.816	0.047	0.862	0.918	0.769	0.026	0.886	0.927	0.835	0.034
HitNet	704*704	0.835	0.889	0.787	0.060	0.857	0.927	0.788	0.025	0.864	0.915	0.818	0.040
Ours	704*704	0.879	0.919	0.832	0.046	0.889	0.935	0.820	0.022	0.893	0.928	0.848	0.033
SAM-Adapter	1024*1024	0.851	0.901	0.781	0.068	0.884	0.917	0.805	0.025	0.887	0.912	0.821	0.040
Ours	1024*1024	0.876	0.916	0.830	0.048	0.895	0.940	0.834	0.021	0.892	0.927	0.849	0.034

potentially aiding in segmenting camouflaged targets and partly because SAM’s local window design is better suited for high-resolution inputs.

Ablation on Multimodal Large Language Models. We investigate the effectiveness of utilizing different MLLMs to generate guiding texts. Results in Tab. 12 show the performance of text semantic information obtained from different MLLMs. Specifically, we train our model with textual captions which is generated by OpenFlamingo [31], IDEFICS [26], and MiniGPT-4 [53], respectively. During textual description extraction, we assign the same instruction to each model. Tab. 11 illustrates examples of text generated by various models. Due to the utilization of stronger pre-trained language models, MiniGPT-4 produces highly diversified text, offering detailed descriptions of images, including camouflaged objects, surrounding environments, habitats, and additional information. Conversely, models equipped with in-context learning capabilities, such as OpenFlamingo and IDEFICS, generate more structured output results. In our

Tab. 11: Demonstration of generated text for camouflaged images with different MLLMs. MiniGPT-4 produces richer textual descriptions, while OpenFlamingo and IDEFICS, which have contextual demonstrations, generate more structured outputs.

Image	Model	Content
	GT	The small size frog is camouflaged in the right of the image.
	OpenFlamingo [31]	The small size frog is camouflaged in the middle of the image.
	IDEFICS [26]	The small size frog is camouflaged in the center of the image.
	MiniGPT-4 [53]	The image shows a small frog sitting on ... It has large eyes and a long, sticky tongue ... The size of the frog is small, ... The category of the frog is amphibian. The location of the frog is in a forest.

Tab. 12: Ablations on using different VLMs to extract textual information for guiding the model at the stage of text semantic generation.

Methods	CAMO (250)				COD10K (2,026)				NC4K (4,121)			
	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$
MiniGPT-4	0.879	0.920	0.831	0.047	0.893	0.937	0.829	0.021	0.888	0.923	0.843	0.036
IDEFICS	0.880	0.919	0.833	0.047	0.896	0.940	0.837	0.020	0.892	0.926	0.848	0.034
OpenFlamingo	0.876	0.916	0.830	0.048	0.895	0.940	0.834	0.021	0.892	0.927	0.849	0.034

approach, OpenFlamingo and IDEFICS achieve competitively close performance.

Employing text generated by MiniGPT-4 as guidance results in lower performance when compared to the other two models. We speculate that excessive text semantics and language model hallucinations, such as mentioning “*a long, sticky tongue*” may lead the trained model astray when dealing with abnormally camouflaged objects.

Ablation on Encoding Methods. In this experiment, we explore the impact of different text encoding methods on the performance of our TG-COD. We utilize both hard encoding and CLIP to encode the generated guiding text. Size, category, and location are encoded as separate one-hot vectors, with the vector length determined by the number of representative words for each semantic information. For instance, we distinguish size of object in image as large, medium, or small. Words not matching any of these are labeled as “other”, resulting in a vector dimension of 4 to represent size. Similarly, we use two vectors to represent category and location, respectively. Subsequently, we concat these three vectors into a single one. The experimental results are presented in Tab.

Tab. 13: Ablations for encoding methods of textual semantics. "AD" means the baseline, which involves fine-tuning without the guidance of text semantics. "Hard Encoder" and "CLIP" represent methods of encoding textual information.

Methods	CAMO (250)				COD10K (2,026)				NC4K (4,121)			
	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$
AD(baseline)	0.861	0.905	0.812	0.055	0.883	0.928	0.817	0.023	0.882	0.916	0.835	0.037
Hard Encode	0.873	0.916	0.817	0.051	0.891	0.937	0.827	0.020	0.889	0.924	0.844	0.036
CLIP	0.876	0.916	0.830	0.048	0.895	0.940	0.834	0.021	0.892	0.927	0.849	0.034

Tab. 14: Ablations for VLM with different numbers of shots at the stage of text semantic generation.

No. of Shot	CAMO (250)				COD10K (2,026)				NC4K (4,121)			
	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$
0	0.866	0.911	0.811	0.053	0.889	0.933	0.824	0.022	0.888	0.921	0.842	0.037
2	0.876	0.917	0.826	0.049	0.896	0.941	0.834	0.021	0.890	0.926	0.847	0.035
4	0.876	0.916	0.830	0.048	0.895	0.940	0.834	0.021	0.892	0.927	0.849	0.034

13. Both encoder methods surpass the baseline, demonstrating the benefits of textual guidance. As MLLM’s text output is open-vocabulary, categories that are morphologically or biologically similar exhibit closer distances in their encoded vectors, reflecting the inherent connections between different semantic categories. However, hard encoding fails to capture this intrinsic relationship. We speculate that this might explain its slightly lower performance.

Ablation on Number of Shot in In-context Learning. Experiments are performed on 0, 2, and 4 shots to validate the efficacy of the demonstration in our In-context learning-based inferences. The results in Tab. 14 shows that 2 and 4 shots outperform 0 shot, demonstrating the contribution of ICL to enhancing the quality of generated text.

Impact of Quality of Generated Text. This experiment explores the effects of generated guiding text of different qualities on TG-COD performance. Due to the presence of blank texts directly outputted by OpenFlamingo, which could affect the model’s performance, we adopt a cropping and multiple-generation approach to enhance the quality of generated guiding texts. In order to demonstrate the effectiveness of the proposed method in the stage of text semantic generation, and to explore the impact of

Tab. 15: Ablation study of different text generation methods. "AD" means the baseline without the text guidance. "Single" represents the direct extraction of a single inference, and "Multiple" corresponds to the best test generation from multiple ones.

Methods	Empty Text	Similarity	COD10K-Base(1473)			
			$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$
AD*	-	-	0.845	0.890	0.744	0.034
Single*	31	0.895	0.853	0.900	0.754	0.032
Multiple*	0	0.907	0.855	0.903	0.760	0.031

text quality on model performance, we design a set of ablation experiments. We regard the performance of the adapter as a baseline. Tab. 15 shows the performance without employing the multiple generation strategy, only using unsupervised demonstration filtering, as well as the performance in the final setup. We replace the empty text with *"This is a camouflaged object image."*. We count the number of blank text samples in the training data and evaluate the similarity between generated texts and ground truth texts using a language model (MiniLM) to assess text quality. It can be observed that our method avoids the interference of blank texts and improves text quality, contributing to the final performance.

Effectiveness of Textual Guidance. To explore the benefits of each key component of text generation under low-data conditions, we conduct ablation experiments under a few-shot setting. We separately experiment using generated guiding text and ground truth text. The results in Tab. 16 indicate that integrating semantic information, which effectively guides visual models to adapt better to COD task, enhances the accuracy of the model's recognition. The presence of inherent noise in generated guiding text may lead to model misinterpretation. Therefore, the performance under the SZ+CLS+LOC* setting does not significantly surpass that of the CLS+LOC* setting. We hypothesize that the marginal variance in text quality between the two settings may account for this outcome. Nevertheless, considering its overall competitive performance, we still regard it as the final configuration. Our generated guiding text outperform the utilization of ground truth category cues to guide the model, suggesting that even amidst some degree of noise, generated text remains a more effective guiding mechanism than

Tab. 16: Our TG-COD’s performance under different textual guidance. ”-” means without text information. ”Fix” means replacing the generated text from the fix text ”This is an image.” ”SZ”, ”CLS”, and ”LOC” means the size, category, location information, respectively.

Methods	COD10K-Base(1473)							
	Generated Text				Ground Truth Text			
	$S_a \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$	$S_a \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$
-*	0.845	0.890	0.744	0.034	-	-	-	-
Fix*	0.850	0.896	0.747	0.033	-	-	-	-
CLS*	<u>0.856</u>	0.902	0.751	0.031	0.856	0.901	0.751	0.031
SZ+LOC*	0.852	0.898	0.753	0.033	0.866	0.925	<u>0.776</u>	<u>0.026</u>
SZ+CLS*	0.855	0.900	<u>0.754</u>	<u>0.031</u>	<u>0.867</u>	0.919	<u>0.755</u>	<u>0.026</u>
CLS+LOC*	0.857	0.904	<u>0.754</u>	0.030	0.863	0.912	0.768	0.028
SZ+CLS+LOC*	0.855	<u>0.903</u>	0.760	<u>0.031</u>	0.869	<u>0.923</u>	0.784	0.025

ground truth category cues. It is noteworthy that guiding the model using ground truth text undoubtedly achieves optimal performance, while the current performance of the model using generated text exhibits a significant gap, revealing that there is still room for further improvement in text semantic guidance methods. Adopting more advanced techniques to enhance the quality of generated text has the potential to bring further improvements. Moreover, the performance under the SZ+LOC setting surpasses that of the CLS and CLS+LOC settings. This implies that spatial information might yield superior results compared to categorical information. We hypothesize that this phenomenon stems from SAM’s capacity to leverage spatial prompts for segmentation, thereby demonstrating its robust spatial discrimination capability.

Ablation on Guiding Text. Considering the potential domain shift between ground truth text and generated text, training the model with ground truth text may induce overfitting to the ground truth data, thus leading to suboptimal performance. To validate this hypothesis, we trained the model separately using both generated and ground truth text, followed by testing using generated text. The results presented in Tab. 17 demonstrate that the model trained on ground truth text yields lower test performance.

Further more, we conducted additional experiments with inaccurate (mismatched) or irrelevant (random) text. We randomly select text from the AG’s news topic classification dataset as irrelevant text and randomly replace information in the generated text

Tab. 17: Training the model using generated text instead of ground truth text aims to mitigate domain shift between the guiding text during training and testing processes. * means training under the few-shot setting. "-Base" ("-Novel") denotes images of classes include (not include) in the few-shot training dataset.

Methods	COD10K-Base(36,360,1276)				COD10K-Novel(33,750)			
	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$
Ground Truth*	0.843	0.890	0.756	0.034	0.819	0.865	0.715	0.036
Generate*	0.855	0.903	0.760	0.031	0.835	0.880	0.722	0.035

Tab. 18: Irrelevant: Randomly selecting text from the AG’s news topic classification dataset. Inaccurate: Randomly replacing information in the generated text (including size, location, and category).

Methods	COD10K (2,026)				NC4K (4,121)			
	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$MAE \downarrow$
irrelevant	0.879	0.926	0.812	0.025	0.874	0.909	0.828	0.041
inaccurate	0.882	0.930	0.820	0.023	0.881	0.915	0.833	0.038
ours	0.895	0.940	0.834	0.021	0.892	0.927	0.849	0.034

as inaccurate text. The results in Tab. 18 indicate that both inaccurate and irrelevant text lead to a decline in performance, even falling below the baseline without text guidance.

5. Conclusion

This work explores how to extract and leverage useful pseudo-textual information from multimodal LLM. Our proposed Text-Guided Camouflaged Object Detection(TG-COD) framework achieves state-of-the-art(SOTA) performance during this exploration and exhibits high generalization to the few-shot setting. How to combine pseudo information from various foundation models remains to study, which we leave to future work.

Appendix A. Visualization

This section presents additional results from ablation experiments aimed at validating the efficacy of text semantic guidance. The visualizations provided in Figs. A.1 and A.2 offer compelling evidence that the proposed method of text guidance enhances model performance in following aspects:

- Natural language descriptions facilitate a more comprehensive understanding of the camouflaged object detection(COD) task by the model, thereby enabling clearer object segmentation. This clarity is particularly beneficial for distinguishing camouflaged objects from other highly confusing background targets.
- Category information implicitly encodes intrinsic knowledge about the morphological features, color, appearance, and other attributes of target animals, thereby assisting in segmenting more complete camouflaged objects.
- Information about the location and size aids in locating and detecting tiny-scale objects within images.

Text	The small size scorpionfish is camouflaged in the top-right of the image.	The small size frogfish is camouflaged in the bottom of the image.	The small size shrimp is camouflaged in the top-left of the image.	The small size ghostpipefish is camouflaged in the bottom-left of the image.	The medium size octopus is camouflaged in the center of the image.	The small size pipefish is camouflaged in the top-right of the image.	The small size pipefish is camouflaged in the top of the image.	The small size starfish is camouflaged in the middle of the image.
Image								
GT								
Adapter								
Ours								
Text	The medium size heron is camouflaged in the center of the image.	The small size frog is camouflaged in the middle of the image.	The small size toad is camouflaged in the right of the image.	The medium size toad is camouflaged in the top-right of the image.	The small size toad is camouflaged in the center of the image.	The small size mantis is camouflaged in the top of the image.	The small size moth is camouflaged in the center of the image.	The small size snake is camouflaged in the top of the image.
Image								
GT								
Adapter								
Ours								
Text	The small size cat is camouflaged in the center of the image.	The small size cat is camouflaged in the bottom-right of the image.	The small size spider is camouflaged in the center of the image.	The medium size spider is camouflaged in the center of the image.	The medium size stickinsect is camouflaged in the top-left of the image.	The small size stickinsect is camouflaged in the bottom-right of the image.	The small size grasshopper is camouflaged in the center of the image.	The small size grasshopper is camouflaged in the center of the image.
Image								
GT								
Adapter								
Ours								

Fig. A.1: Visualization of ablation experiment. "Text" is the generated guiding text. "Adapter" means training the model without the guidance of the generated text. Our TG-COD with text guidance can generate more precise and complete predictions.



Fig. A.2: Visualization of ablation experiment. "Text" is the generated guiding text. "Adapter" means training the model without the guidance of the generated text. Our TG-COD with text guidance can generate more precise and complete predictions.

References

- [1] D.-P. Fan, G.-P. Ji, G. Sun, M.-M. Cheng, J. Shen, L. Shao, Camouflaged object detection, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 2777–2787.
- [2] D.-P. Fan, G.-P. Ji, M.-M. Cheng, L. Shao, Concealed object detection, IEEE transactions on pattern analysis and machine intelligence 44 (10) (2021) 6024–6042, <https://doi.org/10.1109/TPAMI.2021.3085766>.
- [3] M. Türkoğlu, D. Hanbay, Plant disease and pest detection using deep learning-based features, Turkish Journal of Electrical Engineering and Computer Sciences 27 (3) (2019) 1636–1651, <https://doi.org/10.3906/elk-1809-181>.
- [4] D.-P. Fan, G.-P. Ji, T. Zhou, G. Chen, H. Fu, J. Shen, L. Shao, Pranet: Parallel reverse attention network for polyp segmentation, in: International conference on medical image computing and computer-assisted intervention, Springer, 2020, pp. 263–273, https://doi.org/10.1007/978-3-030-59725-2_26.
- [5] D. Tabernik, S. Šela, J. Skvarč, D. Skočaj, Segmentation-based deep-learning approach for surface-defect detection, Journal of Intelligent Manufacturing 31 (3) (2020) 759–776, <https://doi.org/10.1007/s10845-019-01476-x>.
- [6] Y. Pang, X. Zhao, T.-Z. Xiang, L. Zhang, 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, <https://doi.org/10.1109/cvpr52688.2022.00220>.
- [7] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al., On the opportunities and risks of foundation models, arXiv preprint arXiv:2108.07258 (2021).
- [8] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al., Segment anything, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 4015–4026, <https://doi.org/10.1109/ICCV51070.2023.00371>.

- [9] C. He, K. Li, Y. Zhang, G. Xu, L. Tang, Y. Zhang, Z. Guo, X. Li, Weakly-supervised concealed object segmentation with sam-based pseudo labeling and multi-scale feature grouping, *Advances in Neural Information Processing Systems* 36 (2024).
- [10] X.-J. Luo, S. Wang, Z. Wu, C. Sakaridis, Y. Cheng, D.-P. Fan, L. Van Gool, Camdiff: Camouflage image augmentation via diffusion, *CAAI Artificial Intelligence Research* 2 (2023).
- [11] N. Houlsby, A. Giurui, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, S. Gelly, Parameter-efficient transfer learning for nlp, in: *International Conference on Machine Learning*, PMLR, 2019, pp. 2790–2799.
- [12] M. Zhuge, X. Lu, Y. Guo, Z. Cai, S. Chen, Cubenet: X-shape connection for camouflaged object detection, *Pattern Recognition* 127 (2022) 108644, <https://doi.org/10.1016/j.patcog.2022.108644>.
- [13] Q. Jia, S. Yao, Y. Liu, X. Fan, R. Liu, Z. Luo, Segment, magnify and reiterate: Detecting camouflaged objects the hard way, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4713–4722, <https://doi.org/10.1109/CVPR52688.2022.00467>.
- [14] T. Zhou, Y. Zhou, C. Gong, J. Yang, Y. Zhang, Feature aggregation and propagation network for camouflaged object detection, *IEEE Transactions on Image Processing* 31 (2022) 7036–7047, <https://doi.org/10.1109/tip.2022.3217695>.
- [15] G.-P. Ji, L. Zhu, M. Zhuge, K. Fu, Fast camouflaged object detection via edge-based reversible re-calibration network, *Pattern Recognition* 123 (2022) 108414, <https://doi.org/10.1016/j.patcog.2021.108414>.
- [16] H. Mei, G.-P. Ji, Z. Wei, X. Yang, X. Wei, D.-P. Fan, Camouflaged object segmentation with distraction mining, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8772–8781, <https://doi.org/10.1109/cvpr46437.2021.00866>.

- [17] W. Zhai, Y. Cao, H. Xie, Z.-J. Zha, Deep texton-coherence network for camouflaged object detection, *IEEE Transactions on Multimedia*<https://doi.org/10.1109/tmm.2022.3188401> (2022).
- [18] A. Li, J. Zhang, Y. Lv, B. Liu, T. Zhang, Y. Dai, Uncertainty-aware joint salient object and camouflaged object detection, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10071–10081, <https://doi.org/10.1109/cvpr46437.2021.00994>.
- [19] Q. Zhai, X. Li, F. Yang, C. Chen, H. Cheng, D.-P. Fan, Mutual graph learning for camouflaged object detection, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12997–13007, <https://doi.org/10.1109/cvpr46437.2021.01280>.
- [20] Y. Cheng, H.-Z. Hao, Y. Ji, Y. Li, C.-P. Liu, Attention-based neighbor selective aggregation network for camouflaged object detection, in: *2022 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2022, pp. 1–8, <https://doi.org/10.1109/ijcnn55064.2022.989215>.
- [21] H. Zhang, C. Qin, Y. Yin, Y. Fu, Camouflaged image synthesis is all you need to boost camouflaged detection, *arXiv preprint arXiv:2308.06701* (2023).
- [22] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, *Advances in neural information processing systems* 33 (2020) 6840–6851.
- [23] Z. Chen, R. Gao, T.-Z. Xiang, F. Lin, Diffusion model for camouflaged object detection, in: *ECAI 2023*, IOS Press, 2023, pp. 445–452, <https://doi.org/10.3233/FAIA230302>.
- [24] Z. Chen, K. Sun, X. Lin, Camodiffusion: Camouflaged object detection via conditional diffusion models, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38, 2024, pp. 1272–1280, <https://doi.org/10.1609/aaai.v38i2.27890>.
- [25] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, et al., Flamingo: a visual language model for

- few-shot learning, *Advances in Neural Information Processing Systems* 35 (2022) 23716–23736.
- [26] H. Laurençon, L. Saulnier, L. Tronchon, S. Bekman, A. Singh, A. Lozhkov, T. Wang, S. Karamcheti, A. Rush, D. Kiela, et al., Obelics: An open web-scale filtered dataset of interleaved image-text documents, *Advances in Neural Information Processing Systems* 36 (2024).
- [27] J. Hu, J. Lin, S. Gong, W. Cai, 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, 2024, pp. 12511–12518, <https://doi.org/10.1609/aaai.v38i11.29144>.
- [28] T. Chen, L. Zhu, C. Deng, R. Cao, Y. Wang, S. Zhang, Z. Li, L. Sun, Y. Zang, P. Mao, Sam-adapter: Adapting segment anything in underperformed scenes, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3367–3375, <https://doi.org/10.1109/ICCVW60793.2023.00361>.
- [29] Y. Lei, L. Lei, Few-shot object detection with camouflage animals, in: *Journal of Physics: Conference Series*, Vol. 1961, IOP Publishing, 2021, p. 012005, <https://doi.org/10.1088/1742-6596/1961/1/012005>.
- [30] T.-D. Nguyen, A.-K. N. Vu, N.-D. Nguyen, V.-T. Nguyen, T. D. Ngo, T.-T. Do, M.-T. Tran, T. V. Nguyen, The art of camouflage: Few-shot learning for animal detection and segmentation, *IEEE Access* <https://doi.org/10.1109/ACCESS.2024.3432873> (2024).
- [31] A. Awadalla, I. Gao, J. Gardner, J. Hessel, Y. Hanafy, W. Zhu, K. Marathe, Y. Bitton, S. Gadre, S. Sagawa, et al., Openflamingo: An open-source framework for training large autoregressive vision-language models, *arXiv preprint arXiv:2308.01390* (2023).
- [32] Y. N. Dauphin, A. Fan, M. Auli, D. Grangier, Language modeling with gated convolutional networks, in: *International conference on machine learning*, PMLR, 2017, pp. 933–941.

- [33] G.-P. Ji, D.-P. Fan, P. Xu, M.-M. Cheng, B. Zhou, L. Van Gool, Sam struggles in concealed scenes—empirical study on” segment anything”, arXiv preprint arXiv:2304.06022 (2023).
- [34] T.-N. Le, T. V. Nguyen, Z. Nie, M.-T. Tran, A. Sugimoto, Anabran network for camouflaged object segmentation, *Computer vision and image understanding* 184 (2019) 45–56, <https://doi.org/10.1016/j.cviu.2019.04.006>.
- [35] Y. Lv, J. Zhang, Y. Dai, A. Li, B. Liu, N. Barnes, 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. 11591–11601, <https://doi.org/10.1109/cvpr46437.2021.01142>.
- [36] D.-P. Fan, M.-M. Cheng, Y. Liu, T. Li, A. Borji, Structure-measure: A new way to evaluate foreground maps, in: *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4548–4557, <https://doi.org/10.1007/s11263-021-01490-8>.
- [37] D.-P. Fan, C. Gong, Y. Cao, B. Ren, M.-M. Cheng, A. Borji, Enhanced-alignment measure for binary foreground map evaluation, in: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, International Joint Conferences on Artificial Intelligence Organization*, 2018, <https://doi.org/10.24963/ijcai.2018/97>.
- [38] R. Margolin, L. Zelnik-Manor, A. Tal, How to evaluate foreground maps?, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 248–255, <https://doi.org/10.1109/CVPR.2014.39>.
- [39] F. Perazzi, P. Krähenbühl, Y. Pritch, A. Hornung, Saliency filters: Contrast based filtering for salient region detection, in: *2012 IEEE conference on computer vision and pattern recognition*, IEEE, 2012, pp. 733–740, <https://doi.org/10.1109/cvpr.2012.6247743>.
- [40] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from

- natural language supervision, in: International conference on machine learning, PMLR, 2021, pp. 8748–8763.
- [41] H. Zhu, P. Li, H. Xie, X. Yan, D. Liang, D. Chen, M. Wei, J. Qin, I can find you! boundary-guided separated attention network for camouflaged object detection, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, 2022, pp. 3608–3616, <https://doi.org/10.1609/aaai.v36i3.20273>.
 - [42] Y. Sun, S. Wang, C. Chen, T. Z. Xiang, Boundary-guided camouflaged object detection, in: 31st International Joint Conference on Artificial Intelligence, IJCAI 2022, International Joint Conferences on Artificial Intelligence, 2022, pp. 1335–1341, <https://doi.org/10.24963/ijcai.2022/186>.
 - [43] D. Zheng, X. Zheng, L. T. Yang, Y. Gao, C. Zhu, Y. Ruan, Mffn: Multi-view feature fusion network for camouflaged object detection, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2023, pp. 6232–6242, <https://doi.org/10.1109/WACV56688.2023.00617>.
 - [44] X. Hu, S. Wang, X. Qin, H. Dai, W. Ren, D. Luo, Y. Tai, L. Shao, High-resolution iterative feedback network for camouflaged object detection, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, 2023, pp. 881–889, <https://doi.org/10.1609/aaai.v37i1.25167>.
 - [45] Z. Huang, H. Dai, T.-Z. Xiang, S. Wang, H.-X. Chen, J. Qin, 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, <https://doi.org/10.1109/CVPR52729.2023.00538>.
 - [46] W. Liu, X. Shen, C.-M. Pun, X. Cun, Explicit visual prompting for low-level structure segmentations, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 19434–19445, <https://doi.org/10.1109/CVPR52729.2023.01862>.
 - [47] Z. Wu, D. P. Paudel, D.-P. Fan, J. Wang, S. Wang, C. Demonceaux, R. Timofte, L. Van Gool, Source-free depth for object pop-out, in: Proceedings of the

- IEEE/CVF international conference on computer vision, 2023, pp. 1032–1042, <https://doi.org/10.1109/ICCV51070.2023.00101>.
- [48] X. Li, J. Yang, S. Li, J. Lei, J. Zhang, D. Chen, Locate, refine and restore: A progressive enhancement network for camouflaged object detection., in: IJCAI, 2023, pp. 1116–1124, <https://doi.org/10.24963/ijcai.2023/124>.
 - [49] J. Zhang, G. Yang, X. Dai, P. Yang, Spanet: Spatial perceptual activation network for camouflaged object detection, IET Computer Vision <https://doi.org/10.1049/cvi2.12310> (2024).
 - [50] H. Yuan, H. Ge, W. He, C. Huang, Edge optimization detection network for camouflaged targets, in: 2024 6th International Conference on Natural Language Processing (ICNLP), IEEE, 2024, pp. 728–732, <https://doi.org/10.1109/ICNLP60986.2024.10692850>.
 - [51] L. Xu, X. You, F. Jia, K. Liu, Bicod: A camouflaged object detection method directed by cognitive attention, IEEE Sensors Journal 24 (4) (2024) 4711–4721, <https://doi.org/10.1109/JSEN.2023.3343917>.
 - [52] T. H. Phung, H.-J. Chen, H.-H. Shuai, Hierarchically aggregated identification transformer network for camouflaged object detection, in: 2024 IEEE International Conference on Multimedia and Expo (ICME), IEEE, 2024, pp. 1–6, <https://doi.org/10.1109/ICME57554.2024.10687759>.
 - [53] D. Zhu, J. Chen, X. Shen, X. Li, M. Elhoseiny, Minigpt-4: Enhancing vision-language understanding with advanced large language models, in: The Twelfth International Conference on Learning Representations.