Mutual Graph Learning for Camouflaged Object Detection

Qiang Zhai[1,‡] Xin Li[2,‡] Fan Yang[2,*] Chenglizhao Chen[3] Hong Cheng[1] Deng-Ping Fan[4]
[1] UESTC [2] Group 42 (G42) [3] Qingdao University [4] Inception Institute of AI (IIAI)
‡ Equal contributions

Abstract

Automatically detecting/segmenting object(s) that blend in with their surroundings is difficult for current models. A major challenge is that the intrinsic similarities between such foreground objects and background surroundings make the features extracted by deep model indistinguishable. To overcome this challenge, an ideal model should be able to seek valuable, extra clues from the given scene and incorporate them into a joint learning framework for representation co-enhancement. With this inspiration, we design a novel Mutual Graph Learning (MGL) model, which generalizes the idea of conventional mutual learning from regular grids to the graph domain. Specifically, MGL decouples an image into two task-specific feature maps — one for roughly locating the target and the other for accurately capturing its boundary details — and fully exploits the mutual benefits by recurrently reasoning their high-order relations through graphs. Importantly, in contrast to most mutual learning approaches that use a shared function to model all between-task interactions, MGL is equipped with typed functions for handling different complementary relations to maximize information interactions. Experiments on challenging datasets, including CHAMELEON, CAMO and COD10K, demonstrate the effectiveness of our MGL with superior performance to existing state-of-the-art methods. Code is available at https://github.com/fanyang587/MGL.

1. Introduction

Camouflage is an important skill in nature, because it helps certain animals hide from their predators by blending in with their surroundings. The ability of camouflaging, which is closely related to how human perception works, has attracted increasing research attention over past decades. Biological and psychological studies show that it is hard for human beings to quickly spot camouflaged animals or objects [4, 48]. A possible reason is that the primitive function of our visual system may be designed to detect topological properties [2], thus making it difficult to identity camouflaged animals/objects that break up visual edge information of their 'true' bodies. In spite of these biology discoveries, how to make up for this 'flaw' in human perception by Machine is, unfortunately, still an underexplored topic in computer vision.

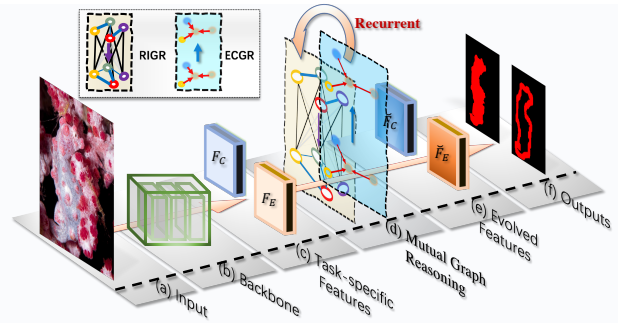


Figure 1: **Illustration of MGL**. Given an image (a), we use a ResNet-FCN as the backbone (b) to extract task-specific features for the camouflaged object detection (COD) and camouflaged object-aware edge extraction (COEE), respectively (c). Then, we exploit the mutual benefits from both tasks by reasoning about their mutual relations with the cooperation of the Region-Induced Graph Reasoning (RIGR) module and Edge-Constricted Graph Reasoning (ECGR) module in a recurrent manner (d). Finally, the evolved features (e) are mapped into the results (f).

Identifying a camouflaged object from its background, also known as camouflaged object detection (COD) [7], is a valuable, yet challenging task [9]. 'Seeing through camouflage' has promising prospects for facilitating various real-life tasks, including image retrieval [29], species discovery [42], traffic risk management, medical image analysis [10, 12, 58], *etc*. However, the existing deep models are still incapable of fully resolving the intrinsic visual similarities between foreground objects and background surroundings. To overcome this difficulty, current approaches distill additional knowledge by extracting auxiliary features from the shared context, *e.g.,* features for identification [9] or classification [20], to significantly augment the underlying

*Corresponding author: Fan Yang *(fanyang_uestc@hotmail.com)*

representations for camouflaged object detection. Although their notable successes truly demonstrate the benefit of exploiting extra knowledge in camouflaged object detection, there are still three major open issues. **First**, the mutual influence between COD and its auxiliary task is overlooked or poorly investigated. More specifically, because the existing efforts [9, 20, 63] only exploit extra information from the auxiliary task to guide/assist the main task (*i.e.* COD), while ignoring the important collaborative relationship between them, these models may fail to a local minimum [49]. **Second**, as the cross-task dependencies are modeled only in the original coordinate space, more global, higher-order guidance information may be lost. As we demonstrate empirically, current COD models become ineffective under heavy occlusions and indefinable boundaries, because they fail to incorporate higher-order information into the representation learning process. **Third**, according to recent biological discoveries [17, 53, 54], a key factor for concealment/camouflage is the *edge disruption*. Unfortunately, how to enhance true edge visibility for facilitating the representation learning for COD is not investigated by existing arts [9, 20], which definitely would weaken, or at least not fully utilize, the COD model's learning power.

Targeting at these drawbacks, we present a novel Mutual Graph Learning model (MGL) to sufficiently and comprehensively exploit mutual benefits between camouflaged object detection (COD) and its auxiliary task. Considering that the *edge disruption* should be one of the key factors for camouflage [17, 53, 54], we treat the camouflaged object-aware edge extraction (COEE) as an auxiliary task and incorporate it into our MGL for mutual learning. As shown in Figure 1, our MGL has a well-designed interweaving architecture that strengthens the interaction and cooperation between tasks. Importantly, instead of 'naïvely' fusing the learned features from two tasks as in the existing works, MGL precisely exploits useful information from the counterparts for representation co-enhancement by explicitly reasoning about the complementary relations between COD and COEE with two typed functions. To mine the semantic guidance information from COD and assist COEE, we develop a novel Region-Induced Graph Reasoning (RIGR) module to reason about the high-level dependencies, and transfer semantic information from COD to augment underlying representations for COEE; To improve the true edge visibility, a new Edge-Constricted Graph Reasoning (ECGR) module is used to explicitly incorporate the edge information from COEE to, in turn, better guide the representation learning for COD. Importantly, our RIGR and ECGR can be formulated in a recurrent manner to recursively mine the mutual benefits and incorporate valuable information from their counterparts.

We demonstrate the effectiveness of our MGL by comparing it against strong baselines and current state-of-the-art methods through extensive experiments on a variety of benchmarks. The experiment results clearly demonstrate its superiority over existing methods in mining mutual guidance information for camouflaged object detection. The contributions of this work are summarized as follows:

- **A novel graph-based, mutual learning approach for camouflaged object detection.** To our knowledge, this is the first attempt to exploit mutual guidance knowledge between two closely related tasks, *i.e.,* COD and COEE, using the graph-based techniques for camouflaged object detection. This approach is able to capture semantic guidance knowledge and spatial supportive information for mutually boosting the performance of both tasks.
- **Carefully designed graph-based interaction functions for fully mining typed guidance information.** Unlike conventional mutual learning approaches, our MGL ensembles two distinct graph-based interaction modules to reason about typed relations: **RIGR** for mining semantic guidance information from COE to assist COEE and, **ECGR** for incorporating true edge priors to enhance the underlying representations of COD.
- **State-of-the-art results on widely-used benchmarks.** Our MGL sets new records on a variety of benchmarks, *i.e., CHAMELEON* [47], *CAMO* [20] and *COD10K* [9], and outperforms existing COD models by a large margin.

2. Related Work

Camouflaged Object Detection. The camouflaged object detection (COD) task [21, 36, 38] has posed new challenges by pushing the boundaries of generic / salient object detection [13, 22–24, 27, 28, 32, 33, 41, 46, 55, 69, 71, 74] to concealed objects blending in with their surroundings. Fan *et al.* [9] present the Search and Identification Net (SINet) to address this challenge by first roughly searching for camouflaged objects and then performing segmentation. Le *et al.* [20] introduce the Anabranch Network (ANet) which incorporates classification information into representation learning. Yan *et al.* [63] introduce MirrorNet to use both instance segmentation and adversarial attack for COD. The common idea behind these bio-inspired models is that exploring and integrating extra clues into representation learning can greatly outperform the conventional approaches for generic object detection (GOD) and salient object detection (SOD) [11, 13, 16, 27, 31, 44, 45, 68, 72]. Unlike prior works, our novelty is that we use a unified, graph-based model to simultaneously perform camouflaged object detection (COD) and the camouflaged object-aware edge extraction (COEE) by comprehensively reasoning about multi-level relations to boost performance for both tasks.

Graph Convolutional Networks. GCNs are powerful tools for graph data analysis, which have given rise to many applications [35, 39, 56, 60, 61, 64, 67]. In the context of (generic/salient) object detection, GCNs are used to detect or segment 2D/3D objects in images, videos or

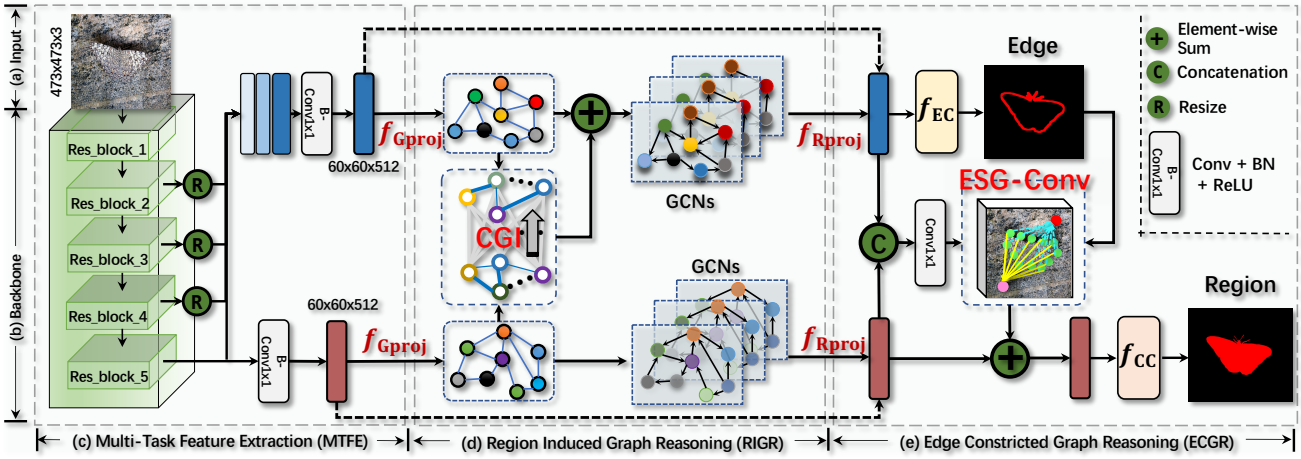


Figure 2: **An overview of our proposed (single-stage) mutual graph learning framework (S-MGL)**. The main components of the flowchart are marked from (a) to (e). **CGI** means the cross-graph interaction module and **ESG-Conv** means the edge supportive graph convolution, which are the key operations for information interactions. Please refer to § 3 for details.

point clouds [50, 52]. In [3, 25], the long-range context is modeled by graph convolution for semantic segmentation. Wu *et al.* [57] exploit the semantic relations and co-occurrence among objects and background with a bidirectional graph. Luo *et al.* [34] introduce a cascade graph model to exploit multi-scale, cross-modality information for salient object detection. In [67], an adaptive GCN model with attention graph clustering is introduced for co-saliency detection. For camouflaged object detection, we introduce two novel graph-based modules, RIGR and ECGR, to fully reason about complementary information of COD and COEE across different levels, which can better learn representations from image to overcome multiple challenges.

3. Our Approach

3.1. Preliminaries

Motivation. Our method is inspired by the discoveries from biological research [17, 53, 54]: capturing the true body/object shape is the key to seeing through camouflage. Then, an ideal model for camouflaged object detection should be well capable of capturing true edges of objects and, more importantly, incorporating such information into a joint learning framework. Intuitively, the involved tasks can benefit each other by information propagation in a unified, graph-based network.

Problem Formulation. Let the COD model be represented by the function $\mathcal{M}_\Theta$ parameterized by weights Θ, that takes an image $\mathbf{I}$ as input, and produces camouflage map $\mathbf{C} \in [0,1]$ and camouflaged object-aware edge map $\mathbf{E} \in [0,1]$ simultaneously, which reflect the probability of each pixel belonging to the camouflaged object(s) and its edges respectively. Our goal is to learn Θ by fully exploiting the mutual benefits between COD and COEE, given the labeled training dataset $\{I_i, C_i, E_i^t\}_{i=1}^N$, where I_i is a training image, C_i means its groundtruth camouflage map, and E_i denotes the true edge map which can be automatically generated from C_i.

3.2. Overview

MGL consists of three major components: Multi-Task Feature Extraction (**MTFE**), Region-Induced Graph Reasoning (**RIGR**) module and Edge-Constricted Graph Reasoning (**ECGR**).

- **MTFE.** Given an input image $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$, a multi-task backbone network $f_{\texttt{MTFE}}$ decouples it into two task-specific representations: $\mathbf{F}_C \in \mathbb{R}^{h\times w\times c}$ for roughly detecting the target and $\mathbf{F}_E \in \mathbb{R}^{h\times w\times c}$ for properly capturing its true edges.
- **RIGR.** In this stage, $\mathbf{F}_C$ and $\mathbf{F}_E$ are first transformed into sample-dependent semantic graphs $\mathcal{G}_C = (\mathcal{V}_C, \mathcal{E}_C)$ and $\mathcal{G}_E = (\mathcal{V}_E, \mathcal{E}_E)$ by the graph projection operation $f_{\texttt{Gproj}}$, where pixels with similar features form a vertex and edges measure the affinity between vertices in a feature space. Then, Cross-Graph Interaction module (CGI) $f_{\texttt{CGI}}$ is used to capture the high-level dependencies between $\mathcal{G}_C$ and $\mathcal{G}_E$ and transfer semantic information from $\mathcal{V}_C$ to $\mathcal{V}_E$: $\mathcal{V}'_E = f_{\texttt{CGI}}(\mathcal{V}_C, \mathcal{V}_E)$. Next, graph reasoning $f_{\texttt{GR}}$ is conducted to obtain evolved graph representations $\mathbf{V}_C$ and $\mathbf{V}'_E$ by graph convolution [18]. At last, $\mathbf{V}_C$ and $\mathbf{V}'_E$ are projected back to the original coordinate space $\hat{\mathbf{F}}_C = f_{\texttt{Rproj}}(\mathbf{V}_C)$ and $\breve{\mathbf{F}}_E = f_{\texttt{Rproj}}(\mathbf{V}'_E)$.
- **ECGR.** Before spatial relationship analysis, $\breve{\mathbf{F}}_E$ is first fed into the edge classifier $f_{\texttt{EC}}$ to obtain camouflaged object-aware edge map $\mathbf{E}$. In addition, we fuse $\breve{\mathbf{F}}_E$ and $\hat{\mathbf{F}}_C$ (*e.g.,* by `concatenate`) to form a new feature map $\mathbf{F}'_C$ for COD, and then use a new Edge Supportive Graph Convolution (`ESG-Conv`) to encode edge information and enhance $\mathbf{F}'_C$ for better locating objects, under

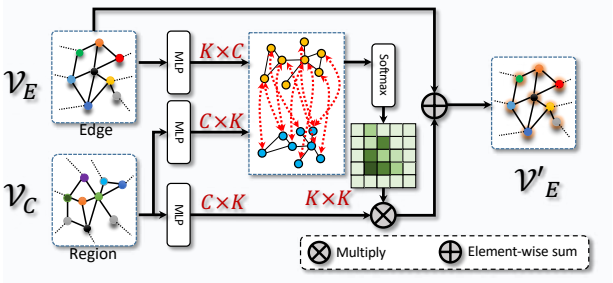


Figure 3: **Illustration of CGI**. CGI promotes the cross-graph (task) interaction, and transfers the information of COD to learn the evolved graph representations for COEE.

the guidance of $\mathbf{E}$: $\breve{\mathbf{F}}_C = \texttt{ESGConv}(\mathbf{F}'_C; \mathcal{G}^e(\mathbf{E}))$ where $\mathcal{G}^e(\mathbf{E})$ denotes the edge supportive graph which is conditioned on $\mathbf{E}$. Finally, we feed $\breve{\mathbf{F}}_C$ into the classifier $f_{\texttt{CC}}$ to obtain the final results $\mathbf{C}$.

Figure 2 presents an overview of our method. In MGL, the mutual relations between COD and COEE are reasoned over multiple levels of interaction spaces by employing two novel neural modules, *i.e.,* RIGR and ECGR. By explicitly reasoning about their relationships, valuable mutual guidance information, intuitively, can be precisely propagated to assist each other during representation learning. It is worth mentioning that RIGR and ECGR can be stacked consecutively for recurrent mutual learning.

3.3. Mutual Graph Learning

Here, we give a detailed introduction to our Multi-Task Feature Extraction (MTFE), Region-Induced Graph Reasoning (RIGR) and Edge-Constricted Graph Reasoning (ECGR).

Multi-Task Feature Extraction (MTFE). $f_{\texttt{MTFE}}$ takes an image as the input, and produces two task-specific feature maps — one for COD and the other for COEE. Formally, given an input image $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$, a multi-task backbone network (*i.e*, a multi-branch ResNet-based FCN network parameterized by $\Theta_{\texttt{MTFE}}$) is employed to simultaneously obtain representations for COD ($\mathbf{F}_C$) and COEE ($\mathbf{F}_E$):

$$\mathbf{F}_C = f_{\texttt{MTFE}}(\mathbf{I}; \Theta_{\texttt{MTFE}}), \quad \mathbf{F}_E = f_{\texttt{MTFE}}(\mathbf{I}; \Theta_{\texttt{MTFE}}), \tag{1}$$

where $\mathbf{F}_C \in \mathbb{R}^{h\times w\times c}$ and $\mathbf{F}_E \in \mathbb{R}^{h\times w\times c}$ are features with $h \times w$ spatial resolution and c channels for COD and COEE respectively, so that spatial information and high-level semantic information can be well preserved.

Region-Induced Graph Reasoning (RIGR). RIGR aims at reasoning about the region-induced semantic relations within COD and between COD and COEE, regardless of local details. It consists of four operations/functions: *(1)* Graph Projection $f_{\texttt{Gproj}}$, *(2)* Cross-Graph Interaction $f_{\texttt{CGI}}$, *(3)* Graph Reasoning $f_{\texttt{GR}}$ and *(4)* Graph Reprojection $f_{\texttt{Rproj}}$.

(1) Graph Projection $f_{\texttt{Gproj}}$. Given input features $\mathbf{F}_C \in \mathbb{R}^{h\times w\times c}$ or $\mathbf{F}_E \in \mathbb{R}^{h\times w\times c}$, we first use a 1×1 convolutional layer to transform them into lower-dimension features, denoted as $\mathbf{F}^l_C \in \mathbb{R}^{(h\times w)\times C}$ or $\mathbf{F}^l_E \in \mathbb{R}^{(h\times w)\times C}$. Then, $f_{\texttt{Gproj}}$ is used to transform feature vectors, $\mathbf{F}^l_C$ or $\mathbf{F}^l_E$, into graph node embeddings/representations, *i.e.,* $\mathcal{V}_C \in \mathbb{R}^{C\times K}$ or $\mathcal{V}_E \in \mathbb{R}^{C\times K}$. Following [26, 66], we parameterize $f_{\texttt{Gproj}}$ by $W \in \mathbb{R}^{K\times C}$ and $\Sigma \in \mathbb{R}^{K\times C}$. Each column w_k of W specifies a learnable clustering center for the k-th node. Specifically, the representation of each node can be computed as follow:

$$v_k = \frac{v'_k}{||v'_k||_2}, \quad v'_k = \frac{1}{\sum_i q^i_k} \sum_i q^i_k (f_i - w_k)/\sigma_k, \tag{2}$$

where σ_k is the column vector of Σ, v'_k is a weighted average of the residuals between feature vector f_i and w_k. v_k means the representation for the k-th node, and forms the k-th column of the node feature matrix $\mathcal{V}$. q^i_k is the soft-assignment of a feature vector f_i to w_k, and can be computed by the following equation:

$$q^i_k = \frac{\exp(-||(f_i - w_k)/\sigma_k||^2_2/2)}{\sum_j \exp(-||(f_i - w_j)/\sigma_j||^2_2/2)}, \tag{3}$$

where '/' means the element-wise division. Here, we compute the graph adjacent matrix by measuring the affinity between intra-node representations: $\mathcal{A}^{\texttt{intra}} = f_{\texttt{norm}}(\mathcal{V}^{\mathrm{T}} \times \mathcal{V}) \in \mathbb{R}^{K\times K}$, where $f_{\texttt{norm}}$ means the normalization operation.

(2) Cross-Graph Interaction $f_{\texttt{CGI}}$. $f_{\texttt{CGI}}$ models the between-graph interaction and guides the inter-graph message passing from $\mathcal{V}_C$ to $\mathcal{V}_E$. This goal leads us to draw inspiration from the non-local operation [51], and compute inter-graph dependencies with attention mechanism. To begin with, as shown in Figure 3, we use different multi-layer perceptrons (MLPs) [43] to transform $\mathcal{V}_C$ to the *key* graph $\mathcal{V}^\theta_C$ and the *value* graph $\mathcal{V}^\gamma_C$, and $\mathcal{V}_E$ to the *query* graph $\mathcal{V}^\kappa_E$. Then, the similarity matrix $\mathcal{A}^{\texttt{inter}}_{C\to E} \in \mathbb{R}^{K\times K}$ is calculated by a matrix multiplication as:

$$\mathcal{A}^{\texttt{inter}}_{C\to E} = f_{\texttt{norm}}({\mathcal{V}^\kappa_E}^{\mathrm{T}} \times \mathcal{V}^\theta_C), \tag{4}$$

where $\mathcal{A}^{\texttt{inter}}_{C\to E} \in \mathbb{R}^{K\times K}$. After that, we can transfer semantic information from $\mathcal{V}_C$ to $\mathcal{V}_E$ by

$$\mathcal{V}'_E = f_{\texttt{CGI}}(\mathcal{V}_C, \mathcal{V}_E) = \chi(\mathcal{A}^{\texttt{inter}}_{C\to E} \times {\mathcal{V}^\gamma_C}^{\mathrm{T}}) + \mathcal{V}_E, \tag{5}$$

where χ acts as the weighting parameter to adjust the importance of CGI *w.r.t.* $\mathcal{V}_E$.

(3) Graph Reasoning $f_{\texttt{GR}}$. After performing inter-graph interaction, we conduct the intra-graph reasoning by taking $\mathcal{V}_C$ and $\mathcal{V}'_E$ as inputs to obtain enhanced graph representations. Here, $f_{\texttt{GR}}$ can be implemented with graph convolution [18]:

$$\begin{cases} \mathbf{V}_C = f_{\texttt{GR}}(\mathcal{V}_C) = g(\mathcal{A}^{\texttt{intra}}_C \mathcal{V}_C W_C) \in \mathbb{R}^{C\times K}, \\ \mathbf{V}'_E = f_{\texttt{GR}}(\mathcal{V}'_E) = g(\mathcal{A}^{\texttt{intra}}_E \mathcal{V}'_E W_E) \in \mathbb{R}^{C\times K}, \end{cases} \tag{6}$$

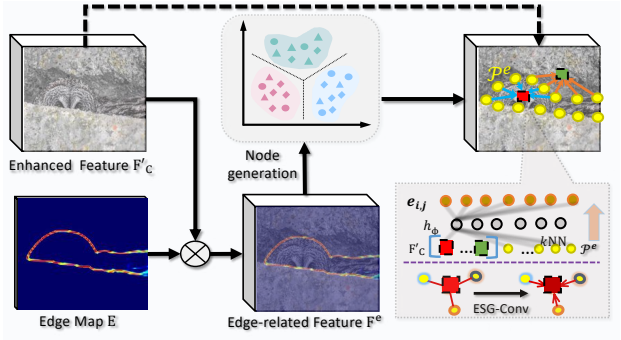


Figure 4: **Illustration of ECGR**. ECGR mines useful information from COEE to guide the representation learning of COD. The supportive nodes are generated in a 'soft' way.

where $g(\cdot)$ is a non-linear activation function, W_C and W_E are learnable parameters of the graph convolution layer, and $\mathcal{A}_C^{\texttt{intra}}$ and $\mathcal{A}_E^{\texttt{intra}}$ denote the graph adjacent matrices for $\mathcal{V}_C$ and $\mathcal{V}'_E$, respectively.

(4) Graph Reprojection $f_{\texttt{Rproj}}$. To map the enhanced graph representations back to the original coordinate space, we revisit the assignments from the graph projection step. Formally, let us denote the assignment matrix for COD as $\mathcal{Q}_C = [q_{Ck}]_{k=0}^{(K-1)}$, where $q_{Ck} = [q_{Ck}^i]_{i=0}^{(h\times w)-1}$, and the assignment matrix for COEE as $\mathcal{Q}_E = [q_{Ek}]_{k=0}^{(K-1)}$, where $q_{Ek} = [q_{Ek}^i]_{i=0}^{(h\times w)-1}$. The graph reprojection $f_{\texttt{Rproj}}$ can be formulated as:

$$\begin{cases} \hat{\mathbf{F}}_C &= \mathcal{Q}_C \mathbf{V}_C^{\mathrm{T}} + \mathbf{F}_C^l, \ \ \mathcal{Q}_C \in \mathbb{R}^{(h\times w)\times K}, \\ \breve{\mathbf{F}}_E &= \mathcal{Q}_E \mathbf{V'}_E^{\mathrm{T}} + \mathbf{F}_E^l, \ \ \mathcal{Q}_E \in \mathbb{R}^{(h\times w)\times K}, \end{cases} \tag{7}$$

where $\hat{\mathbf{F}}_C \in \mathbb{R}^{(h\times w)\times C}$ and $\breve{\mathbf{F}}_E \in \mathbb{R}^{(h\times w)\times C}$ are the enhanced feature maps for COD and COEE respectively.

Edge-Constricted Graph Reasoning (ECGR). ECGR focuses on edge-constricted relation reasoning in order to extract useful information from COEE to further guide the representation learning for COD. The idea illustration for our ECGR is given in Figure 4.

(1) Our Goal. The goal of ECGR is to equip the model with an explicit edge perception capability so as to locate objects accurately. We expect $\hat{\mathbf{F}}_C$ to be updated by explicitly perceiving and encoding information about edge. With this goal, we first produce the enhanced feature map $\mathbf{F}'_C$ for COD by directly fusing $\breve{\mathbf{F}}_E$ and $\hat{\mathbf{F}}_C$ (via `concatenate`), and then use a novel Edge Supportive Graph Convolution (`ESG-Conv`) to update it, conditioned on $\mathbf{E}$. Next, we describe the edge supportive graph $\mathcal{G}^e(\mathbf{E})$ and the graph convolution `ESG-Conv` intended for it.

(2) Supportive Node/Vertex Generation. The first step for building $\mathcal{G}^e(\mathbf{E})$ is to generate edge-based node embeddings. First, we map $\hat{\mathbf{F}}_E$ to a camouflage object-aware edge map $\mathbf{E} \in \mathbb{R}^{h\times w\times 1}$ via a fully connected layer. Then, as shown in Figure 4, we obtain the edge-related features on regular grids of $\mathbf{F}'_C$ in a 'soft' manner with the attention mechanism: $\mathbf{F}^e = \mathbf{E} \otimes \mathbf{F}'_C$, where $\otimes$ means the channel-wise multiplication operation. Finally, a graph projection operation $f_{\texttt{Gproj}}$ is used to transform $\mathbf{F}^e$ into z edge-based node embeddings, denoted as $\mathcal{P}^e = \{p_1^e, \cdots, p_z^e\}$, to represent the edge prior.

(3) Edge Supportive Graph Convolution `ESG-Conv`. We construct our edge supportive graph $\mathcal{G}^e(\mathbf{E}) = (\mathcal{V}^e, \mathcal{E}^e)$ as the k-nearest neighbor (k-NN) graph [52] to link $\mathbf{F'}_C$ with $\mathcal{P}^e$, where $\mathcal{V}^e$ and $\mathcal{E}^e$ denote the vertices and edges respectively. Formally, we regard each feature vector $f_i'^c \in \mathbf{F'}_C$ as the central node and $\{p_j^c : (i,j) \in \mathcal{E}^e\}$ as its edge supportive nodes. The edge embedding $\mathbf{e}_{i,j}$ can be defined as:

$$\mathbf{e}_{i,j} = h_\phi(f_i'^c, p_j^c) = f_{\texttt{Conv}}(f_i'^c - p_j^c), \tag{8}$$

where h_ϕ is a nonlinear function with learnable parameters ϕ. The output of `ESG-Conv` for the i-th feature vector/vertex is thus given as:

$$\breve{\mathbf{f}}_i = \max_{j:(i,j)\in\mathcal{E}^e} h_\Phi(f_i'^c, \mathbf{e}_{i,j}), \tag{9}$$

where h_Φ denotes the function for learning node embeddings with learnable parameters Φ, and $\breve{\mathbf{f}}_i \in \breve{\mathbf{F}}_C$ means the evolved representation. With our `ESG-Conv`, edge information can be explicitly encoded into underlying representations, *i.e.*, $\breve{\mathbf{F}}_C = \texttt{ESGConv}(\mathbf{F}'_C; \mathcal{G}^e(\mathbf{E}))$.

Recurrent Learning Process. To fully exploit the mutual benefits between COD and COEE, we can further formulate our MGL as the following recurrent learning process:

$$\begin{cases} \breve{\mathbf{F}}_E^{(t+1)} = f_{\texttt{RIGR}}(\breve{\mathbf{F}}_C^{(t)}, \breve{\mathbf{F}}_E^{(t)}), \\ \breve{\mathbf{F}}_C^{(t+1)} = f_{\texttt{ECGR}}(\breve{\mathbf{F}}_C^{(t)}, \breve{\mathbf{F}}_E^{(t+1)}, \mathbf{E}^{(t+1)}), \end{cases} \tag{10}$$

where $f_{\texttt{RIGR}}$ and $f_{\texttt{ECGR}}$ means **RIGR** and **ECGR** modules respectively. Note that at the beginning ($t = 1$), $\breve{\mathbf{F}}_C^{(1)} = f_{\texttt{MTFE}}(\mathbf{I}; \Theta_{\texttt{MTFE}})$ and $\breve{\mathbf{F}}_E^{(1)} = f_{\texttt{MTFE}}(\mathbf{I}; \Theta_{\texttt{MTFE}})$.

3.4. Implementation Details

We present two versions of MGL. One, named as **S-MGL**, is a single-stage model which mines the mutual information only once. The other, named as **R-MGL**, includes a recurrent learning process performing two recurrent stages. The implementation is detailed as follows:

Multi-Task Feature Extractor. Following existing arts [9], we employ ResNet-50 [14] pre-trained on ImageNet [19] as the backbone. We use the dilated network technique [65] to ensure that the feature map for COD ($\mathbf{F}_C$) is 60×60 in resolution. To extract features for COEE ($\mathbf{F}_E$), we first collect a set of side-output features $\{\mathbf{S}_k\}_{k=2}^5$ from ResNet-50, then make these features have the same resolution of 60×60 via a bi-linear up/down-sampling layer, and finally fuse them with a `concatenate` layer followed by a 1×1 convolutional layer.

Table 1: **Quantitative results on different datasets**. '†' means SOTA methods for GOD and SOD. ↑ (or ↓) indicates that the higher (or the lower) the better. Online benchmark: http://dpfan.net/camouflage.

Methods	CHAMELEON [47]				CAMO-Test [20]				COD10K-Test [9]			
	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$
2017 FPN † [27]	0.794	0.783	0.590	0.075	0.684	0.677	0.483	0.131	0.697	0.691	0.411	0.075
2017 MaskRCNN † [13]	0.643	0.778	0.518	0.099	0.574	0.715	0.430	0.151	0.613	0.748	0.402	0.080
2017 PSPNet † [68]	0.773	0.758	0.555	0.085	0.663	0.659	0.455	0.139	0.678	0.680	0.377	0.080
2018 UNet++ † [73]	0.695	0.762	0.501	0.094	0.599	0.653	0.392	0.149	0.623	0.672	0.350	0.086
2018 PiCANet † [31]	0.769	0.749	0.536	0.085	0.609	0.584	0.356	0.156	0.649	0.643	0.322	0.090
2019 MSRCNN † [16]	0.637	0.686	0.443	0.091	0.617	0.669	0.454	0.133	0.641	0.706	0.419	0.073
2019 PoolNet † [30]	0.776	0.779	0.555	0.081	0.702	0.698	0.494	0.129	0.705	0.713	0.416	0.074
2019 BASNet † [45]	0.687	0.721	0.474	0.118	0.618	0.661	0.413	0.159	0.634	0.678	0.365	0.105
2019 PFANet † [70]	0.679	0.648	0.378	0.144	0.659	0.622	0.391	0.172	0.636	0.618	0.286	0.128
2019 CPD † [59]	0.853	0.866	0.706	0.052	0.726	0.729	0.550	0.115	0.747	0.770	0.508	0.059
2019 HTC † [1]	0.517	0.489	0.204	0.129	0.476	0.442	0.174	0.172	0.548	0.520	0.221	0.088
2019 EGNet † [69]	0.848	0.870	0.702	0.050	0.732	0.768	0.583	0.104	0.737	0.779	0.509	0.056
2019 ANet-SRM [20]	‡	‡	‡	‡	0.682	0.685	0.484	0.126	‡	‡	‡	‡
2020 MirrorNet [63]	‡	‡	‡	‡	0.741	0.804	0.652	0.100	‡	‡	‡	‡
2020 PraNet [10]	0.860	0.898	0.763	0.044	0.769	0.833	0.663	0.094	0.789	0.839	0.629	0.045
2020 SINet [9]	0.869	0.891	0.740	0.044	0.751	0.771	0.606	0.100	0.771	0.806	0.551	0.051
S-MGL (ours)	0.892	0.921	0.803	0.032	0.772	**0.850**	0.664	0.089	0.811	0.851	0.655	0.037
R-MGL (ours)	**0.893**	**0.923**	**0.813**	**0.030**	**0.775**	0.847	**0.673**	**0.088**	**0.814**	**0.865**	**0.666**	**0.035**

Region-Induced Graph Reasoning Module. We follow [26] to design and implement $f_{\texttt{Gproj}}$, and encode $\mathbf{F}_C$ and $\mathbf{F}_E$ to $K = 32$ semantic nodes respectively (see Table 4). The transformation function in $f_{\texttt{CGI}}$ is implemented by MLPs (1×1 convolution). In our RIGR, Eq. 4 is used to build the *between-graph* relations, and Eq. 5 is used to capture semantic guidance information (from $\mathcal{V}_C$ to $\mathcal{V}_E$) and produce the evolved graph representation $\mathbf{V}'_E$ for $\mathcal{V}'_E$. $f_{\texttt{GR}}$ is implemented via GCNs [18] and $f_{\texttt{Rproj}}$ reuses the assignment matrix for graph re-projection by using Eq. 7.

Edge-Constricted Graph Reasoning Module. For the number of edge supportive nodes, we observe that $z = 32$ can ensure a promising speed-accuracy tradeoff (see Table 4). $h_\phi(\cdot)$ in Eq. 8 can be simply implemented with element-wise *subtraction* operation followed by a 1×1 convolution. $h_\Phi(\cdot)$ in Eq. 9 concatenates edge and node embeddings, *i.e.,* $f_i'^c$ & $\mathbf{e}_{i,j}$, and uses a 1×1 convolution to fuse them for producing $\breve{\mathbf{f}}_i \in \breve{\mathbf{F}}_C$.

Classifier and Loss Function. After obtaining the evolved representations $\breve{\mathbf{F}}_E^{(t)}$ and $\breve{\mathbf{F}}_C^{(t)}$, we use classifiers to map them to the corresponding outputs $\mathbf{E}$ and $\mathbf{C}$, which are implemented by 1×1 convolutional layers. For training, we use bi-linear interpolation to upsample the output maps to the original size to calculate the loss. We use the cross-entropy loss [5] for both tasks:

$$L = L_{CE}^c(\mathbf{C}, \mathbf{G}_C) + \gamma L_{CE}^e(\mathbf{E}, \mathbf{G}_E), \tag{11}$$

where $\mathbf{G}_C$ and $\mathbf{G}_E$ mean the groundtruth labels, and γ means the combination weight. Here we simply set $\gamma = 1$.

4. Experiments

4.1. Experimental Setup

Datasets: We perform extensive experiments on the following public benchmarks:

- **CHAMELEON [47]** includes 76 high-resolution images finely annotated with pixel-level labels. All images in CHAMELEON are collected from the Internet.
- **CAMO [20]** is a collection of $2,500$ images with 8 categories. In this dataset, both naturally camouflaged objects and artificially camouflaged objects are collected with finely-annotated labels.
- **COD10K [9]** is the largest COD dataset, which includes $10,000$ images with 10 super-classes and 78 sub-classes. All images are collected from photography websites.

Our `train set` is a combination of the train sets from CAMO and COD10K provided by [7].

Evaluation Metric: Following [9, 20], we adopt mean absolute error (MAE) as evaluation metric. In addition, mean E-measure (E_ϕ) [8], S-measure (S_α) [6] and weighted F-measure (F_β^w) [37] are used for balanced comparisons. Moreover, for evaluating our auxiliary COEE task, we adopt the precision-recall metric with F-measure following [62]. Evaluation tools: https://github.com/DengPingFan/CODToolbox.

Training Settings: During training, the weights of MTFE are initialized by ResNet-50 [14] pre-trained on ImageNet [19], and the remaining layers/modules are randomly initialized. For data preparation, we perform data augmentation techniques on all training data, including random cropping, left-right flipping and scaling in the range of $[0.75, 1.25]$. For optimization, we use the Stochastic Gradient Descent (SGD) with 'poly' learning rate scheduling policy: $lr = base\_lr \times (1 - \frac{iter}{max_{iter}})^{power}$. The base learning rate $base\_lr$ is set to 10^{-7} and $power$ to 0.9.

Reproducibility: Our S-MGL and R-MGL are implemented based on PyTorch. Our model is trained on a NVIDIA Tesla V100 GPU to ensure a larger batch size. During test, all models are performed on a NVIDIA GTX Titan X GPU with 12G memory.

Table 2: **Ablation study** of the proposed approach on CHAMELEON, CAMO `test` and COD10K `test`.

Candidate				CHAMELEON [47]				CAMO-Test [20]				COD10K-Test [9]			
ResNet-50	RIGR	ECGR	RL	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$
✔				0.767	0.799	0.535	0.094	0.742	0.786	0.538	0.130	0.729	0.692	0.436	0.079
✔	✔			0.844	0.863	0.686	0.055	0.766	0.828	0.611	0.104	0.785	0.758	0.557	0.052
✔	✔	✔		0.892	0.921	0.803	0.032	0.772	**0.850**	0.664	0.089	0.811	0.851	0.655	0.037
✔	✔	✔	✔	**0.893**	**0.923**	**0.813**	**0.030**	**0.775**	0.847	**0.673**	**0.088**	**0.814**	**0.865**	**0.666**	**0.035**

Table 3: **Quantitative results** of different underlying feature enhancement algorithms.

Method	CAMO-Test [20]				COD10K-Test [9]			
	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$
Baseline (ResNet-50 FCN)	0.742	0.786	0.538	0.130	0.729	0.692	0.436	0.079
Baseline + NL [51]	0.748	0.791	0.541	0.122	0.731	0.711	0.459	0.073
MTFE + MUL [40]	0.751	0.799	0.551	0.118	0.736	0.721	0.498	0.070
S-MGL (ours)	0.772	**0.850**	0.664	0.089	0.811	0.851	0.655	0.037
R-MGL (ours)	**0.775**	0.847	**0.673**	**0.088**	**0.814**	**0.865**	**0.666**	**0.035**

Table 4: **Detailed ablation study** of different parameter settings. 'K' means the number of semantic nodes; 'z' stands for the number of edge supportive nodes; 't' means that t recurrent stages are used in our MGL.

Method	CAMO-Test [20]				COD10K-Test [9]			
	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$
S-MGL (K=16, z=32)	0.771	0.832	0.661	0.092	0.805	0.832	0.638	0.042
S-MGL (K=32, z=32)	0.772	**0.850**	0.664	0.089	0.811	0.851	0.655	0.037
S-MGL (K=64, z=32)	0.774	0.849	0.661	0.089	0.809	0.854	0.648	0.037
S-MGL (K=32, z=16)	0.772	0.843	0.662	0.090	0.804	0.837	0.640	0.040
S-MGL (K=32, z=32)	0.772	**0.850**	0.664	0.089	0.811	0.851	0.655	0.037
S-MGL (K=32, z=64)	0.773	0.848	0.666	0.089	0.807	0.855	0.657	0.037
R-MGL (K=32, z=32, t=1)	0.772	**0.850**	0.664	0.089	0.811	0.851	0.655	0.037
R-MGL (K=32, z=32, t=2)	**0.775**	0.847	**0.673**	**0.088**	0.814	**0.865**	**0.666**	**0.035**
R-MGL (K=32, z=32, t=3)	0.773	0.848	0.672	**0.088**	**0.815**	0.862	0.661	0.036

4.2. Comparison with State-of-the-Arts

Baselines / SOTAs: Similar to [9], we first select strong baseline models which achieve SOTA performance in closely related fields, *i.e,* GOD and SOD. Moreover, all recently published methods for COD are included for comparisons. In sum, we compare our methods (S-MGL and R-MGL) against 16 SOTAs, which are trained under their recommended settings with the same `train` set as ours.

Performance on CHAMELEON: Table 1 reports the comparison results with 14 SOTAs on CHAMELEON. For fair comparison, all models use the same `train` set for training. As can be seen, our S-MGL achieves better performance than all compared works across all metrics. When compared with the state-of-the-art SINet [9], S-MGL significantly lowers MAE by 27.3% and improve F_β^w by 8.5%. Our R-MGL further boosts the performance and sets a new record. Clearly, our solution can significantly overcome the ambiguity in camouflaged scenes and provide more reliable results than existing approaches.

Performance on CAMO: We also compare our methods with SOTAs on CAMO `test`. As can be seen in Table 1, our S-MGL and R-MGL achieve significantly better performance than other solutions. This is because our model can fully exploit mutual benefits and ensure model's reliability to overcome the heavy occlusions and indefinable boundaries in complex scenes.

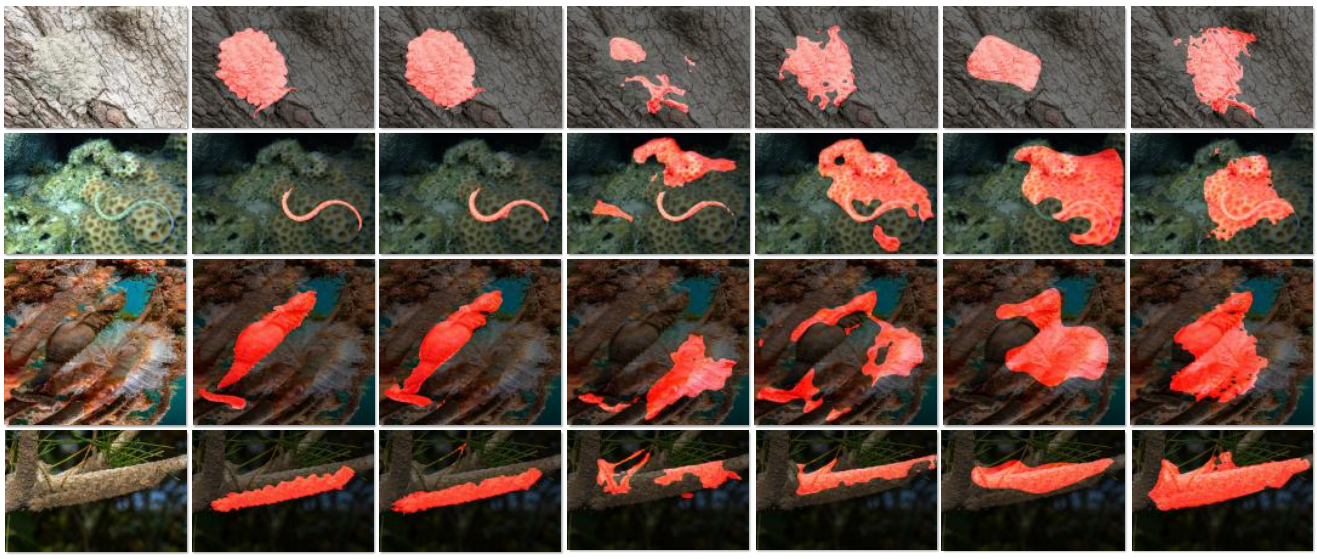
Performance on COD10K: On the largest COD10K `test`, our solution sets new records for all metrics. Specifically, S-MGL greatly surpasses currently best models, which achieves S_α score of 81.1%, E_ϕ score of 85.1%, F_β^w score of 65.5%, and sets the best MAE score of 0.037. R-MGL further boosts the performance. The powerful graph-based interaction modules enable our models to work well with the auxiliary COEE for overcoming all challenges in COD. Some visual samples are given in Figure 5.

Auxiliary Task (COEE): We believe that the mutual learning within our model can also significantly benefit the auxiliary COEE. To verify this, we compare our MGL with the well-known HED [62] and its improved version DSS [15]. Moreover, we include the strong multi-task baseline MUL [40] for comparison. All models are trained on the same `train` set with our extracted edge labels. As shown in Table 5, our S-MGL and R-MGL achieve stronger results than existing models in this task, which shows that our solution can not only improve the performance of the main task (COD) but also boost the auxiliary task (COEE). Some visual samples are provided in Figure 6.

4.3. Ablation Study

Effectiveness of RIGR and ECGR: To verify the effect of our RIGR, we use a model based on ResNet50-FCN as the baseline. First, as shown in Table 2, RIGR enables the model to achieve a certain performance improvement compared to the baseline across all datasets, which demonstrates the effectiveness of the proposed RIGR. Besides, by adding ECGR, we can see a further improvement in accuracy. Thus, it is clear that improving the true edge visibility is important and can empower the model with stronger capability for overcoming difficulties in COD tasks. Moreover, we have carefully studied the parameters in our RIGR and ECGR modules. Table 4 provides the detailed comparisons of different settings.

Usefulness of Recurrent Learning: We can easily extend our MGL into a more comprehensive recurrent reasoning process. Table 2 shows that model's performance can be further improved with recurrent learning techniques. This is because the recurrent process can be used to refine the initial results / features, and thus improve the accuracy. Furthermore, according to our experiments (see Table 4), using only two recurrent steps can ensure promising performance, which makes our R-MGL set new records for all benchmarks and greatly outperform existing approaches.



(a) Image (b) GT (c) MGL (ours) (d) SINet (e) EGNet (f) POOLNet (g) PFANet

Figure 5: **Qualitative comparisons** between different models: (c) our approach (R-MGL), (d) SINet [9], (e) EGNet [69], (f) POOLNet [30], and (g) PFANet [70]. Clearly, our approach can better spot hidden objects with more clear boundaries.

Table 5: **The comparison of camouflaged object-aware edge results** with some wide-used methods on CAMO `test` and COD10K `test`.

Method	CAMO-Test [20]		COD10K-Test [9]	
	ODS	OIS	ODS	OIS
HED [62]	0.315	0.318	0.294	0.313
DSS [15]	0.316	0.336	0.347	0.372
Res50-FCN	0.509	0.511	0.505	0.524
MTEF + MUL [40]	0.521	0.539	0.516	0.534
S-MGL	0.536	0.545	0.535	0.557
R-MGL	**0.543**	**0.551**	**0.540**	**0.558**

Superiority of Mutual Graph Learning: We conduct comprehensive experiments / comparisons to show the superiority of our mutual graph learning approach. As shown in Table 3, compared with the widely used non-local (NL) operation, the explicit mutual learning (MUL) can guarantee more reliable results, which demonstrates that mining the valuable auxiliary edge information can help the model overcome COD challenges, such as heavy occlusions and indefinable boundaries. Our idea is to extend MUL from regular grids to graph domain. Clearly, our S-MGL and R-MGL outperform conventional MUL due to its stronger capability for capturing high-order relations. These experiments demonstrate that deeply mining high-order relations between COD and auxiliary COEE is meaningful, which can significantly improve the reliability of model to better overcome the intrinsic ambiguity for the challenging COD task. Moreover, reasoning high-order relations through graphs would bring clear performance improvements.

5. Conclusion

We have presented the Mutual Graph Learning (MGL), a graph-based, joint learning framework for detecting camouflaged objects and their true edges. Our model includes two novel neural modules: Region-Induced Graph Reasoning (RIGR) module and Edge-Constricted Graph Reasoning (ECGR) module, which can work together to mine valuable complementary information for improving the true edge visibility for COD. We also formulate our MGL as a recurrent graph reasoning process to fully exploit all useful information. Extensive experiments show that explicitly mining true edge prior / information can help to overcome the intrinsic difficulties in COD tasks, such as occlusions and indefinable boundaries. We believe our MGL can also benefit other related computer vision tasks, *e.g.,* panoptic segmentation, that require multi-source information for the joint representation enhancement.

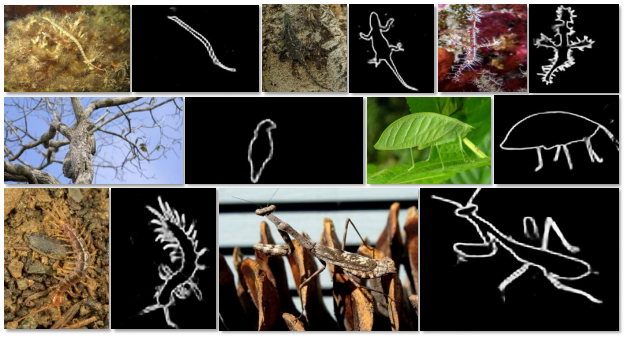


Figure 6: **Visual results** of our approach (R-MGL) for camouflaged object-aware edge extraction on COD10K `test`.

Acknowledgement. This research was funded in part by the National Natural Science Foundation of China (NO. U1964203) and the National Key R&D Program Project of China (2017YFB0102603).

References

[1] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *CVPR*, 2019. 6

[2] Lin Chen. Topological structure in visual perception. *Science*, 1982. 1

[3] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Yan Shuicheng, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. In *CVPR*, June 2019. 3

[4] IC Cuthill. Camouflage. *Journal of Zoology*, 2019. 1

[5] Pieter-Tjerk De Boer, Dirk P Kroese, Shie Mannor, and Reuven Y Rubinstein. A tutorial on the cross-entropy method. *Annals OR*, 2005. 6

[6] Deng-Ping Fan, Ming-Ming Cheng, Yun Liu, Tao Li, and Ali Borji. Structure-measure: A new way to evaluate foreground maps. In *ICCV*, 2017. 6

[7] Deng-Ping Fan, Ge-Peng Ji, Ming-Ming Cheng, and Ling Shao. Concealed object detection. *arXiv preprint arXiv:2102.10274*, 2021. 1, 6

[8] Deng-Ping Fan, Ge-Peng Ji, Xuebin Qin, and Ming-Ming Cheng. Cognitive vision inspired object segmentation metric and loss function. *SSI*, 2021. 6

[9] Deng-Ping Fan, Ge-Peng Ji, Guolei Sun, Ming-Ming Cheng, Jianbing Shen, and Ling Shao. Camouflaged object detection. In *CVPR*, 2020. 1, 2, 5, 6, 7, 8

[10] Deng-Ping Fan, Ge-Peng Ji, Tao Zhou, Geng Chen, Huazhu Fu, Jianbing Shen, and Ling Shao. Pranet: Parallel reverse attention network for polyp segmentation. In *MICCAI*, 2020. 1, 6

[11] Deng-Ping Fan, Tengpeng Li, Zheng Lin, Ge-Peng Ji, Dingwen Zhang, Ming-Ming Cheng, Huazhu Fu, and Jianbing Shen. Re-thinking co-salient object detection. *IEEE TPAMI*, 2021. 2

[12] Deng-Ping Fan, Tao Zhou, Ge-Peng Ji, Yi Zhou, Geng Chen, Huazhu Fu, Jianbing Shen, and Ling Shao. Inf-net: Automatic covid-19 lung infection segmentation from ct images. *IEEE TMI*, 39(8):2626 – 2637, 2020. 1

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 2, 6

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5, 6

[15] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip HS Torr. Deeply supervised salient object detection with short connections. In *CVPR*, 2017. 7, 8

[16] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring r-cnn. In *CVPR*, 2019. 2, 6

[17] Changku Kang, Martin Stevens, Jong-yeol Moon, Sang-Im Lee, and Piotr G Jablonski. Camouflage through behavior in moths: the role of background matching and disruptive coloration. *Behavioral Ecology*, 2015. 2, 3

[18] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. 2017. 3, 4, 6

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 5, 6

[20] Trung-Nghia Le, Tam V. Nguyen, Zhongliang Nie, Minh-Triet Tran, and Akihiro Sugimoto. Anabranch network for camouflaged object segmentation. *CVIU*, 2019. 1, 2, 6, 7, 8

[21] Aixuan Li, Jing Zhang, Yunqiu Lyu, Bowen Liu, Tong Zhang, and Yuchao Dai. Uncertainty-aware joint salient object and camouflaged object detection. In *CVPR*, 2021. 2

[22] Xin Li, Fan Yang, Leiting Chen, and Hongbin Cai. Saliency transfer: An example-based method for salient object detection. In *IJCAI*, 2016. 2

[23] Xin Li, Fan Yang, Hong Cheng, Junyu Chen, Yuxiao Guo, and Leiting Chen. Multi-scale cascade network for salient object detection. In *ACM MM*, 2017. 2

[24] Xin Li, Fan Yang, Hong Cheng, Wei Liu, and Dinggang Shen. Contour knowledge transfer for salient object detection. In *ECCV*, 2018. 2

[25] Xia Li, Yibo Yang, Qijie Zhao, Tiancheng Shen, Zhouchen Lin, and Hong Liu. Spatial pyramid based graph reasoning for semantic segmentation. In *CVPR*, 2020. 3

[26] Yin Li and Abhinav Gupta. Beyond grids: Learning graph representations for visual recognition. In *NeurIPS*, pages 9225–9235, 2018. 4, 6

[27] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 2, 6

[28] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 2

[29] Guanghai Liu and Dengping Fan. A model of visual attention for natural image retrieval. In *ICISCCC*, 2013. 1

[30] Jiang-Jiang Liu, Qibin Hou, Ming-Ming Cheng, Jiashi Feng, and Jianmin Jiang. A simple pooling-based design for real-time salient object detection. In *CVPR*, 2019. 6, 8

[31] Nian Liu, Junwei Han, and Ming-Hsuan Yang. Picanet: Learning pixel-wise contextual attention for saliency detection. In *CVPR*, 2018. 2, 6

[32] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016. 2

[33] Ao Luo, Xin Li, Fan Yang, Zhicheng Jiao, and Hong Cheng. Webly-supervised learning for salient object detection. *Pattern Recognition*, 2020. 2

[34] Ao Luo, Xin Li, Fan Yang, Zhicheng Jiao, Hong Cheng, and Siwei Lyu. Cascade graph neural networks for rgb-d salient object detection. In *ECCV*, 2020. 3

[35] Ao Luo, Fan Yang, Xin Li, Dong Nie, Zhicheng Jiao, Shangchen Zhou, and Hong Cheng. Hybrid graph neural networks for crowd counting. In *AAAI*, 2020. 2

[36] Yunqiu Lyu, Jing Zhang, Yuchao Dai, Li Aixuan, Bowen Liu, Nick Barnes, and Deng-Ping Fan. Simultaneously localize, segment and rank the camouflaged objects. In *CVPR*, 2021. 2

[37] Ran Margolin, Lihi Zelnik-Manor, and Ayellet Tal. How to evaluate foreground maps? In *CVPR*, 2014. 6

[38] Haiyang Mei, Ge-Peng Ji, Ziqi Wei, Xin Yang, Xiaopeng Wei, and Deng-Ping Fan. Camouflaged object segmentation with distraction mining. In *CVPR*, 2021. 2

[39] Abduallah Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *CVPR*, 2020. 2

[40] Xuecheng Nie, Jiashi Feng, and Shuicheng Yan. Mutual learning to adapt for joint human parsing and pose estimation. In *ECCV*, 2018. 7, 8

[41] Youwei Pang, Xiaoqi Zhao, Lihe Zhang, and Huchuan Lu. Multi-scale interactive network for salient object detection. In *CVPR*, 2020. 2

[42] Ricardo Pérez-de la Fuente, Xavier Delclòs, Enrique Peñalver, Mariela Speranza, Jacek Wierzchos, Carmen Ascaso, and Michael S Engel. Early evolution and ecology of camouflage in insects. *PNAS*, 2012. 1

[43] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 4

[44] Xuebin Qin, Deng-Ping Fan, Chenyang Huang, Cyril Diagne, Zichen Zhang, Adrià Cabeza Sant'Anna, Albert Suàrez, Martin Jagersand, and Ling Shao. Boundary-aware segmentation network for mobile and web applications. *arXiv preprint arXiv:2101.04704*, 2021. 2

[45] Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jagersand. Basnet: Boundary-aware salient object detection. In *CVPR*, 2019. 2, 6

[46] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 2

[47] P Skurowski, H Abdulameer, J Błaszczyk, T Depta, A Kornacki, and P Kozieł. Animal camouflage analysis: Chameleon database. *Unpublished Manuscript*, 2018. 2, 6, 7

[48] Martin Stevens and Sami Merilaita. Animal camouflage: current issues and new perspectives. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 2009. 1

[49] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 2

[50] Lei Wang, Yuchun Huang, Yaolin Hou, Shenman Zhang, and Jie Shan. Graph attention convolution for point cloud semantic segmentation. In *CVPR*, 2019. 3

[51] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 4, 7

[52] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *TOG*, 2019. 3, 5

[53] Richard J Webster. Does disruptive camouflage conceal edges and features? *Current Zoology*, 2015. 2, 3

[54] Richard J Webster, Christopher Hassall, Chris M Herdman, Jean-Guy J Godin, and Thomas N Sherratt. Disruptive camouflage impairs object recognition. *Biology Letters*, 2013. 2, 3

[55] Jun Wei, Shuhui Wang, Zhe Wu, Chi Su, Qingming Huang, and Qi Tian. Label decoupling framework for salient object detection. In *CVPR*, 2020. 2

[56] Xin Wei, Ruixuan Yu, and Jian Sun. View-gcn: View-based graph convolutional network for 3d shape analysis. In *CVPR*, 2020. 2

[57] Yangxin Wu, Gengwei Zhang, Yiming Gao, Xiajun Deng, Ke Gong, Xiaodan Liang, and Liang Lin. Bidirectional graph reasoning network for panoptic segmentation. In *CVPR*, 2020. 3

[58] Yu-Huan Wu, Shang-Hua Gao, Jie Mei, Jun Xu, Deng-Ping Fan, Rong-Guo Zhang, and Ming-Ming Cheng. Jcs: An explainable covid-19 diagnosis system by joint classification and segmentation. *IEEE TIP*, 30:3113–3126, 2021. 1

[59] Zhe Wu, Li Su, and Qingming Huang. Cascaded partial decoder for fast and accurate salient object detection. In *CVPR*, 2019. 6

[60] Guo-Sen Xie, Jie Liu, Huan Xiong, and Ling Shao. Scale-aware graph neural network for few-shot semantic segmentation. In *CVPR*, 2021. 2

[61] Guo-Sen Xie, Li Liu, Fan Zhu, Fang Zhao, Zheng Zhang, Yazhou Yao, Jie Qin, and Ling Shao. Region graph embedding network for zero-shot learning. In *ECCV*, 2020. 2

[62] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *ICCV*, 2015. 6, 7, 8

[63] Jinnan Yan, Trung-Nghia Le, Khanh-Duy Nguyen, Minh-Triet Tran, Thanh-Toan Do, and Tam V Nguyen. Mirrornet: Bio-inspired adversarial attack for camouflaged object segmentation. *arXiv*, 2020. 2, 6

[64] Han Yang, Xingjian Zhen, Ying Chi, Lei Zhang, and Xian-Sheng Hua. Cpr-gcn: Conditional partial-residual graph convolutional network in automated anatomical labeling of coronary arteries. In *CVPR*, 2020. 2

[65] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. 2015. 5

[66] Hang Zhang, Jia Xue, and Kristin Dana. Deep ten: Texture encoding network. In *CVPR*, 2017. 4

[67] Yaobin Zhang, Weihong Deng, Mei Wang, Jiani Hu, Xian Li, Dongyue Zhao, and Dongchao Wen. Global-local gcn: Large-scale label noise cleansing for face recognition. In *CVPR*, 2020. 2, 3

[68] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 2, 6

[69] Jia-Xing Zhao, Jiang-Jiang Liu, Deng-Ping Fan, Yang Cao, Jufeng Yang, and Ming-Ming Cheng. Egnet: Edge guidance network for salient object detection. In *ICCV*, 2019. 2, 6, 8

[70] Ting Zhao and Xiangqian Wu. Pyramid feature attention network for saliency detection. In *CVPR*, 2019. 6, 8

[71] Xiaoqi Zhao, Youwei Pang, Lihe Zhang, Huchuan Lu, and Lei Zhang. Suppress and balance: A simple gated network for salient object detection. In *ECCV*, 2020. 2

[72] Huajun Zhou, Xiaohua Xie, Jian-Huang Lai, Zixuan Chen, and Lingxiao Yang. Interactive two-stream decoder for accurate and fast saliency detection. In *CVPR*, 2020. 2

[73] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net ar-

chitecture for medical image segmentation. In *DLMIA*, pages 3–11, 2018. 6

[74] Mingchen Zhuge, Deng-Ping Fan, Nian Liu, Dingwen Zhang, Dong Xu, and Ling Shao. Salient object detection via integrity learning. *arXiv preprint arXiv:2101.07663*, 2021. 2