Global-Local Aggregation with Deformable Point Sampling for Camouflaged Object Detection

Minhyeok Lee\textsuperscript{1} Suhwan Cho\textsuperscript{1} Chaewon Park\textsuperscript{1} Dogyoon Lee\textsuperscript{1} Jungho Lee\textsuperscript{1} Sangyoun Lee\textsuperscript{1,2}

\textsuperscript{1} Yonsei University
\textsuperscript{2} Korea Institute of Science and Technology (KIST)

Abstract

The camouflaged object detection (COD) task aims to find and segment objects that have a color or texture that is very similar to that of the background. Despite the difficulties of the task, COD is attracting attention in medical, lifesaving, and anti-military fields. To overcome the difficulties of COD, we propose a novel global-local aggregation architecture with a deformable point sampling method. Further, we propose a global-local aggregation transformer that integrates an object’s global information, background, and boundary local information, which is important in COD tasks. The proposed transformer obtains global information from feature channels and effectively extracts important local information from the subdivided patch using the deformable point sampling method. Accordingly, the model effectively integrates global and local information for camouflaged objects and also shows that important boundary information in COD can be efficiently utilized. Our method is evaluated on three popular datasets and achieves state-of-the-art performance. We prove the effectiveness of the proposed method through comparative experiments.

1. Introduction

Camouflaged object detection (COD) aims to segment objects that are difficult for humans to detect owing to their similar appearance to the background. Recently, COD has been used in various applications such as polyp segmentation in the medical field, lifesaving in extreme environments, and antimilitary camouflage. COD is similar to salient object detection (SOD) in that there is no prior information about the target object; however, it is a more challenging task because the target is camouflaged. Therefore, sensitive detection of the colors and patterns of camouflaged objects that are intrinsically different from the background is important for COD.

To solve these difficulties, various studies \cite{10,17,18,31,34,35,47,48} suggest deep learning-based methods. Some methods \cite{31,48} extract the global texture feature of the camouflaged object to learn its information separated from the background. However, for most camouflaged objects, the texture is similar to that of the background; therefore, learning subtle differences in boundary local information is a challenging task. In addition, Jia et al. \cite{18} employs a multistage prediction method to achieve high performance. It predicts the camouflaged object again after magnifying the image based on its pre-predicted position. However, this method has a disadvantage in that the inference time is large because the same model is used repeatedly. Several recent studies \cite{17,35,47} improve model performance by applying an additional module that reconstructs boundaries from encoder features. Nevertheless, because these methods extract boundary information directly from the encoder, the performance of the model depends on the quality of the encoder features.

To solve these problems, we propose a novel global-local aggregation with a deformable point sampling network (GLaDOS). The proposed model uses the global and local aggregation transformer (GLA transformer) module to effectively extract and aggregate global and local features.
from a scene. This transformer identifies the overall tendency of the image by considering the global context and local details of the image simultaneously. Moreover, it helps in understanding object boundaries by capturing the fine structure of the object and consequently allows the model to generate an accurate prediction map.

As shown in Figure 1, the GLA transformer is divided into the global extractor, local extractor, and aggregator. First, the global extractor creates soft object region masks for the objects and the background from the input features. Global features are extracted from each feature map area by applying global average pooling to the generated mask region. This process enables effective extraction of global features of camouflaged objects and specifying their approximate location.

The local extractor aims to learn the detailed local information of camouflaged objects and the characteristics that distinguish them from the background. Accordingly, the input feature map is split into patches and local features are extracted from each patch region. We also apply a novel deformable point sampling method to reduce the computational amount and effectively sample useful pixel information within the patch. The proposed sampling method samples features into a more meaningful area for COD through the offset predicted by the offset encoder. In addition, it reduces the amount of computation of self-attention operation of the transformer by calculating the offset for a reference point that is less than the number of input pixels. This method is covered in detail in Section 3.3.

Finally, the aggregator aggregates the generated global and local features. The aggregated information is passed to the boundary decoder for boundary reconstruction of the camouflaged object, and the reconstructed boundary map is used to generate an accurate prediction mask.

We tested our method on three popular datasets: CHAMELEON [32], CAMO [21], and COD10K [10]. These datasets contain various challenging scenarios, and the proposed model achieves state-of-the-art performance on all three datasets. Additionally, we demonstrate the effectiveness of the proposed method through various ablation studies.

2. Related Work

2.1. Salient Object Detection

The SOD task detects and segments objects that visually attract the most human interest from an image. Recent SOD methods [7, 22, 29, 44] have achieved outstanding performance. However, despite recent advances in deep learning, existing SOD methods are very challenging in COD, due to camouflaged objects, multiple objects, transparent objects, and extreme lighting conditions.

2.2. Camouflaged Object Detection

Traditional COD methods [2, 11, 16, 19] use hand-crafted features such as color, boundary, texture, convex, and brightness to distinguish camouflaged objects from their backgrounds. However, these hand-based methods have poor detection results when the target object has colors and textures that are very similar to those of the background.

To solve this problem, deep learning-based COD methods [10, 17, 18, 31, 34, 35, 46–48] have been recently proposed. For example, SINet [10] approaches COD as an SOD problem and applies sophisticated SOD technology to the network. Some studies [31, 48] extract texture information to separate the target object from the background. However, these methods often fail when the target object shares the same texture features with the background. Jia et al. [18] uses a network to specify the approximate location of the target object and repeats the process of magnifying and cropping the image based on this approximate location. However, their iterative multi-stage method has the disadvantage of significantly increasing the model’s inference time. Zhong et al. [46] proposes a frequency-enhancement module to focus on feature points in the frequency domain that are distinct from the background. Furthermore, several recent papers [17, 35, 42, 47] improve the model’s performance by reconstructing the boundaries of the target object from encoder features and integrating them. However, because boundary information is directly extracted from the encoder, there is a problem that the quality of the reconstructed boundary is dependent on the quality of the encoder feature.
3. Proposed Approach

3.1. Overall Architecture

Figure 2 shows the overall architecture of GLaDOS. The proposed model comprises an RGB encoder, a boundary decoder for boundary reconstruction, and a feature-pyramid-network-based decoder for generating a final segmentation map. As a first step, the proposed model extracts and integrates multi-scale features from encoder blocks using multi-scale feature fusion modules (MFFMs). Next, the proposed GLA transformer block extracts global and local features of target objects and aggregates those features. The boundary decoder predicts the boundary map $E_{pred}$, and $E_{pred}$ is merged with the aggregated features through the boundary fusion modules (BFMs). Finally, the model generates a final prediction mask $I_{pred}$ through a decoder.

3.2. Multi-Scale Feature Fusion Module

We use MFFM for effectively extracting and integrating multi-scale features from encoders. Inspired by atrous spatial pyramid pooling [3], MFFM comprises dilated convolutional layers with different ratios as shown in Figure 3. In particular, we apply $3 \times 3$ dilated convolutions with ratios of 6, 12, and 18, respectively. Further, encoder features are sequentially integrated from the small to the large ratio. This structure effectively increases the receptive field with a small number of parameters and delivers rich multi-scale contextual information to the GLA transformers, which will be described later.

3.3. Global-Local Aggregation Transformer

The GLA transformer aims to effectively extract global and local features from MFFM-generated features and aggregate them. As shown in Figure 4, the proposed GLA transformer is mainly composed of a global extractor, a local extractor, an aggregator, and a correlation map generator.

Global Extractor. The global extractor generates global templates $T_g$ from the input feature $X_{in}$, where $C$, $H$, and $W$ indicate the channel, height, and width of the input feature, respectively. As a first step, the global extractor separates each channel of the input feature to generate a channel-separated feature, as shown in Figure 4 (a). In particular, the i-th channel of $X_{in}$ is defined as $X^i_{in} \in \mathbb{R}^{C \times H \times W}$. Next, a channel-wise softmax operation is applied to create soft object regions [41] $X^i_{s} \in [0, 1]^{H \times W}$. According to [41], each channel in $X_{in}$ is generated from the convolutional kernels of the trained encoder, $X_{s}$ contains approximate areas for background or foreground objects. Then, to generate global features, a global weighted
average pooling (GWAP) [30] operation is performed with \( X_{\text{in}} \), treating \( X_{\text{sa}} \) as a weighted mask. In other words, the global template \( t^i_g \in \mathbb{R}^C \) generated by \( X^i_s \) is expressed as follows:

\[
\begin{align*}
  t^i_g &= \text{GAP} \left( X_{\text{in}} \cdot X^i_s \right),
\end{align*}
\]

where \( i = 1, 2, \ldots, C \), and \( \text{GAP} \cdot \cdot \) is a global average pooling operator. Finally, the global extractor creates a global template block \( T_g \), which is a set of generated global templates \( t^i_g \). This method observes the image as a whole and extracts representative features of the scene based on the semantic context. Therefore, \( T_g \) contains global information and thus includes prior knowledge to distinguish objects or backgrounds. Because the number of channels in \( T_g \) is \( C \), the size of \( T_g \) is \( C \times C \).

**Local Extractor.** The local extractor separates the input features into patches and extracts the local features for each patch. As shown in Figure 4 (b), the local extractor splits the input feature \( X_{\text{in}} \) into \( N_p \times N_p \) patches. Therefore, the size of the \( i \)-th patch \( x^i_p \) is \( C \times \frac{H_p}{N_p} \times \frac{W_p}{N_p} \). However, like many vision transformer methods [6,33,39,45], extracting features from every pixel in each patch and applying a transformer is computationally expensive and makes model convergence difficult. To solve this problem, we propose a local feature extraction method inspired by deformable attention [49]. First, \( N_r \times N_r \) reference points are initialized uniformly on \( x^i_p \) as shown in the Figure 4 (b). Next, \( x^i_p \) is feed into a small offset encoder \( \theta_{\text{off}} \) to create an offset field. \( \theta_{\text{off}} \) includes two convolutional layers and one GeLU [14] layer between them, and the tangent hyperbolic function (tanh) is used as the output activation function of \( \theta_{\text{off}} \). In addition, a predefined factor \( s \) is applied to prevent drastic movement of reference points and stabilize learning. In other words, the offset field \( f_o \in (-s, +s)^2 \times N_r \times N_r \) is expressed as follows:

\[
\begin{align*}
  f_o &= s \times \text{tanh} \left( \theta_{\text{off}} \left( x^i_p \right) \right),
\end{align*}
\]

where \( f_o \) represents the relative amount of change in the x- and y-axis directions of each reference point, where \( x^i_p \) is regarded to have a size of \( 1 \times 1 \). Finally, each reference point is moved according to \( f_o \) and the features of size \( C \times 1 \times 1 \) corresponding to that pixel are sampled. However, as exactly locating the moved reference points on a specific pixel on \( x^i_p \) is impossible, we follow [49] to sample the feature by applying bilinear interpolation to 4 adjacent pixels. Local templates generated from \( x^i_p \) are defined as \( t^i_p \). Further, the local template \( T_l \) generated from the entire patches is a set of \( t^1_p, t^2_p, \ldots, t^{(N_p \times N_p)} \). Therefore, as \( N_r \times N_r \) local templates are sampled for one patch, the size of \( T_l \) sampled from the entire patch is \( C \times (N_p \times N_p) \times (N_r \times N_r) \). Consequently, \( T_l \) contains local features extracted from each patch. Moreover, instead of storing information for every pixel, only key features are stored according to deformable attention for computational efficiency.

![Figure 4. Structure of the GLA transformer, composed primarily of four subparts. (a) The global extractor generates global templates of input features. (b) The local extractor separates the input features into patches and extracts the local templates from each patch. (c) The aggregator aggregates the extracted global and local templates. (d) The correlation map generator generates correlation maps from the aggregated features.](image-url)
Aggregator. The aggregator aims to generate useful features for camouflaged object mask reconstruction by effectively aggregating the extracted global templates $T_g$ and local templates $T_1$. Therefore, for a given module, considering the relations between global and local templates, between patches in local extractors, and between local templates within patches is important. Therefore, we design an aggregator inspired by CurveNet [38] in the 3D point cloud classification task. As shown in Figure 4 (c), first key, query, and value-based multi-head attention [12,36,43] are applied to enhance the correlation between $T_g$ templates and the correlation between $T_1$ templates. Next, an attentive pooling (AP) [15] operation is applied along each axis of $T_1$ to generate an inter-patch local feature $T_{1\text{inter}}^{i} \in \mathbb{R}^{C \times (N_r \times N_r)}$ and an intra-patch local feature $T_{1\text{intra}}^{i} \in \mathbb{R}^{C \times (N_p \times N_p)}$. Further, $T_g$, $T_{1\text{inter}}^{i}$, and $T_{1\text{intra}}^{i}$ are fed to individual multi-layer perceptrons (MLPs). In addition, as shown in Figure 4 (c), matrix multiplication and softmax operation are applied among $T_g$, $T_{1\text{inter}}^{i}$, and $T_{1\text{intra}}^{i}$ to generate two correlation score maps $S_{\text{inter}}^{i} \in (0, 1)^{C \times (N_r \times N_r)}$ and $S_{\text{intra}}^{i} \in (0, 1)^{C \times (N_p \times N_p)}$. In another branch, $T_{1\text{inter}}^{i}$ and $T_{1\text{intra}}^{i}$ are further transformed with two extra MLPs, which are then fused with the correlation score maps by matrix multiplication separately. With the above process, two types of aggregated features—$T_{a\text{inter}}^{i} \in \mathbb{R}^{C \times C}$ and $T_{a\text{intra}}^{i} \in \mathbb{R}^{C \times C}$—are created; finally, $T_a \in \mathbb{R}^{C \times C}$ are fused by the MLP layer.

Correlation Map Generator. The correlation map generator generates correlation features from $T_a$. Each template of $T_a$ is treated as a $1 \times 1$ convolution kernel and convolution is performed with $X_{in}$. Because $T_a$ contains a total of $C$ templates, the size of the generated correlation feature $C_a$ is $C \times H \times W$. Finally, $C_a$ and $X_{in}$ are concatenated and the final output feature $X_a \in \mathbb{R}^{C \times H \times W}$ is generated through $1 \times 1$ convolution.

3.4. Boundary Decoder

Figure 5 (a) shows the structure of the proposed boundary decoder. The boundary decoder aims to effectively extract the boundary of camouflaged objects from aggregated multi-scale features $X_1^a$, $X_2^a$, $X_3^a$, and $X_4^a$ from the GLA transformers. The boundary decoder comprises a $3 \times 3$ convolution layer and upsampling layers, integrating the multi-scale features. Finally, all the features are concatenated and passed to the convolutional and sigmoid layers to create a single-channel boundary map $E_{pred}$.

3.5. Boundary Fusion Module

We propose the BFM to integrate the features from the GLA transformers and boundary map with different levels of feature representation using boundary information as a guide. As shown in Figure 5 (b), the input of the proposed BFM comprises the boundary map $E_{pred}$ generated by the boundary decoder and the feature $X_1^a$ generated by the $i$-th GLA transformer, both of which are resized to the same size. Next, $E_{pred}$ and $X_1^a$ are multiplied to generate $X_m$, which is passed it through the GWAP and MLP layers to generate the attention vector $f_a$. Then, $f_a$ and $X_m$ are multiplied to extract boundary-guided global context information. Finally, BFM generates the boundary-fused feature $X_1^f$ by summing boundary-guided global context information and $X_1^a$ and applying $1 \times 1$ convolution.

3.6. Objective Function

Two types of supervision are applied: a camouflaged object mask $L_{co}$ and a boundary map $L_b$. First, weighted binary cross-entropy loss $L_{BCE}^{co}$ and weighted IOU loss $L_{IOU}^{co}$ are applied to $L_{co}$, inspired by the works of [5,37], which helps assign more weight to the hard case pixels. In addition, binary cross-entropy loss $L_{BCE}$ is applied to $L_b$, and a $1 \times 1$ dilation kernel is employed for the ground truth boundary map to solve the lack of supervision signal due to the thin boundary map. Thus, the final objective function $L_{total}$ is expressed as follows:

$$L_{total} = L_{co} (E_{pred}, I_{gt}) + L_b (E_{pred}, E_{gt}) .$$  \hspace{1cm} (3)
4. Experiments

4.1. Datasets

We perform experiments on three popular COD benchmarks to validate the effectiveness of the proposed method: CHAMELEON [32], CAMO [21], and COD10K [10]. CHAMELEON [32] is a small dataset containing only 76 images, which are collected from the Internet. The CAMO [21] dataset includes 1250 images (1000 images in the train set and 250 images in the test set). Finally, COD10K [10] is the largest dataset, containing 10000 images with 10 super-classes and 78 sub-classes collected from websites. Following the method of [10, 27, 28, 35, 47], we use images with camouflaged objects in the experiments, in which 3040 images from COD10K [10] and 1000 images from CAMO [21] are used for training, and the remaining images are employed for testing.
Table 1. Performance comparison of the proposed method with other state-of-the-art methods on the CAMO [21], COD10K [10], and CHAMELEON [32] datasets. ↑ indicates that higher is better, and ↓ indicates that lower is better. The best and second-best performances are highlighted in red and blue, respectively.

| Method          | Year    | Backbone      | Size  | CAMO [21] | COD10K [10] | CHAMELEON [32] |
|-----------------|---------|---------------|-------|-----------|-------------|----------------|
|                 |         |               |       | S_e | E_F | ξ | M | ↓ | S_e | E_F | ξ | M | ↓ | S_e | E_F | ξ | M | ↓ |
| SINet [19]      | CVPR 2020 | ResNet50     | 352 x 352 | 0.751 | 0.771 | 0.606 | 0.100 | 0.751 | 0.806 | 0.551 | 0.051 | 0.809 | 0.891 | 0.740 | 0.044 |
| TANet [31]      | TCSVT 2021 | ResNeXt50   | 384 x 384 | 0.793 | 0.834 | 0.690 | 0.083 | 0.803 | 0.848 | 0.629 | 0.041 | 0.888 | 0.911 | 0.786 | 0.036 |
| TiNet [48]      | AAAI 2021 | ResNet50     | 352 x 352 | 0.781 | 0.847 | 0.678 | 0.087 | 0.793 | 0.848 | 0.635 | 0.043 | 0.874 | 0.916 | 0.783 | 0.038 |
| C2F-Net [34]    | IJCAI 2021 | ResNet50    | 352 x 352 | 0.796 | 0.854 | 0.719 | 0.080 | 0.813 | 0.890 | 0.686 | 0.036 | 0.888 | 0.935 | 0.828 | 0.032 |
| PNet [38]       | CVPR 2021 | ResNet50     | 416 x 416 | 0.782 | 0.852 | 0.695 | 0.085 | 0.800 | 0.868 | 0.660 | 0.040 | 0.882 | 0.942 | 0.810 | 0.033 |
| R-MGL [42]      | CVPR 2021 | ResNet50     | -        | 0.775 | 0.847 | 0.673 | 0.088 | 0.814 | 0.865 | 0.666 | 0.035 | 0.893 | 0.923 | 0.813 | 0.030 |
| Rank-Net [27]   | CVPR 2021 | ResNet50     | 352 x 352 | 0.708 | 0.755 | 0.645 | 0.105 | 0.760 | 0.831 | 0.658 | 0.045 | 0.842 | 0.896 | 0.794 | 0.046 |
| Joint-COD [23]  | CVPR 2021 | ResNet50     | 352 x 352 | 0.803 | 0.853 | -    | 0.076 | 0.817 | 0.892 | -    | 0.035 | 0.894 | 0.943 | 0.810 | 0.030 |
| UGTR [40]       | ICCV 2021 | ResNet50     | -        | 0.785 | 0.859 | 0.686 | 0.086 | 0.818 | 0.850 | 0.667 | 0.035 | 0.888 | 0.918 | 0.796 | 0.031 |
| ERRNet [17]     | PR 2022  | ResNet50     | 352 x 352 | 0.761 | 0.817 | 0.660 | 0.088 | 0.780 | 0.867 | 0.629 | 0.044 | 0.877 | 0.927 | 0.805 | 0.036 |
| CANet [25]      | WACV 2022 | ResNet50     | 480 x 480 | 0.807 | 0.866 | 0.767 | 0.075 | 0.832 | 0.890 | 0.745 | 0.032 | 0.901 | 0.940 | 0.843 | 0.028 |
| BSA-Net [47]    | AAAI 2022 | ResNeXt50    | 384 x 384 | 0.796 | 0.851 | 0.717 | 0.079 | 0.818 | 0.891 | 0.699 | 0.034 | 0.895 | 0.946 | 0.841 | 0.027 |
| SegMar [18]     | CVPR 2021 | ResNet50     | 352 x 352 | 0.815 | 0.872 | 0.742 | 0.071 | 0.833 | 0.895 | 0.724 | 0.033 | 0.906 | 0.954 | 0.860 | 0.025 |
| BGNet [35]      | IJCAI 2021 | ResNeXt50   | 416 x 416 | 0.812 | 0.870 | 0.749 | 0.073 | 0.831 | 0.901 | 0.722 | 0.033 | -    | -    | -    | -    |
| Ours            | ResNeXt50 | 352 x 352    | 0.816 | 0.882 | 0.775 | 0.073 | 0.818 | 0.897 | 0.751 | 0.031 | 0.902 | 0.960 | 0.879 | 0.024 |

4.2. Evaluation Metrics

We evaluate the performance of our method by employing four evaluation metrics: mean absolute error (MAE, M), S-measure ($S_e$) [8], E-measure ($E_F$) [9], and weight F-measure ($F_w$) [1].

4.3. Implementation Details

The patch size of the proposed model and the number of reference points are set to 12 x 12 and 3 x 3, respectively. We implement the proposed method using the open deep-learning framework PyTorch. The backbone network is Res2Net50 [13], pre-trained with ImageNet dataset [4]. All the input images are resized to 352 x 352 and augmented by randomly horizontal flipping. During the training stage, we used the Adam optimizer [20] with $w_1 = 0.9$, $w_2 = 0.999$, and $\epsilon = 10^{-8}$. The learning rate decayed from $10^{-4}$ to $10^{-5}$ with the cosine annealing scheduler [26]. Further, we set the total number of epochs to 100 with a batch size 16. Two NVIDIA RTX 2080 Ti GPUs are used for all experiments in this study.

4.4. Comparison with State-of-the-Art Methods

Quantitative Results. Tables 1 shows the quantitative results of the proposed GLaDOS. In general, a large image size shows good test performance in segmentation tasks. We evaluate the proposed model with an image size 352 x 352, the smallest test image size known from previous COD tasks. Nevertheless, the proposed method achieves state-of-the-art performance on all the three challenging datasets. It also outperforms traditional methods in almost all evaluation metrics. In particular, compared with BSA-Net [47] and BGNet [35], which use the same backbone encoder and a similar boundary-reconstruction method, the proposed model achieves higher performance despite the smaller test image size. This shows that the proposed method can extract information about the disguised object more effectively than the existing boundary-reconstruction method. We demonstrate the effectiveness of the proposed modules through various ablation studies in Section 4.5.

Qualitative Results. In Figure 6, we compare our qualitative results with those of five state-of-the-art COD approaches, BGNet [35], BSA-Net [47], Joint-COD [23], Rank-Net [27], and SINet [10], in several challenging scenarios, including multiple objects, low contrast, thin objects, and long distance. As shown in Figure 6 (b) and (c), when the target object has a texture very similar to that of the background, the proposed model produces an accurate prediction map. Furthermore, the proposed method is robust to scenes with multiple common camouflaged objects as in Figure 6 (h) and (i). This is because the proposed GLA transformer learns the relationship between patches to enhance long-distance connectivity and the aggregator efficiently integrates them. In addition, even when thin and long objects are included as in Figure 6 (f) and (i), information advantageous for generating prediction maps such as edges can be extracted with the proposed deformable point sampling method. The usefulness of the deformable point sampling methods will be discussed in detail in Section 4.5.

4.5. Ablation Analysis

We verify the performance of our model through various ablation studies. Table 2 presents the effects of the proposed modules in various combinations. In particular, the baseline refers to a model comprising a simple encoder and decoder. Effect of MFFM. As revealed by (a) and (b) in Table 2, the use of the proposed MFFM improves the performance compared to the baseline model. This is because the parallel dilated convolution integrates multi-scale contextual features and increases the size of the receptive field.
Additionally we describe the result of sampling all pixels.

Figure 7. Comparison of the reconstructed boundary map between the baseline model and the model applied with the proposed GLA transformers.

**Effect of GLA Transformer.** In Table 2, (d) and (g) present the effect of the proposed GLA transformer. Similar to (d), unlike the existing methods of reconstructing the boundary map from encoder features, in (g), the GLA transformer is located between the encoder and boundary decoder. The GLA transformer shows a significant improvement in COD performance on all datasets. Additionally, we compare the results of the model without and with the GLA transformer to show that the GLA transformer can help reconstruct the boundary map. Figure 7 shows the results of the reconstructed boundary map with and without the GLA transformer (models in (b) and (e) of Table 2). Models without GLA transformers produce very sparse boundary maps from images with extremely uncertain boundaries, which act as noise in the final prediction mask. However, applying a GLA transformer produces precise boundary map predictions with the same boundary decoder. This is strong evidence that the proposed GLA transformer delivers additional rich information about the camouflaged object to the boundary decoder.

**Effect of Deformable Point Sampling Method.** Figure 8 describes the results of additional experiments on the deformable point sampling method of the local extractor of the GLA transformer. The number of sampled reference points is changed to $0 \times 0$, $1 \times 1$, $2 \times 2$, $3 \times 3$, $4 \times 4$, $6 \times 6$. Additionally we describe the result of sampling all pixels.

As shown in Figure 8, when using the deformable point sampling method, the performance generally increases as the number of sampled pixels increases, but does not show a significant increase beyond $3 \times 3$. Furthermore, COD performance is improved by using the deformable method without fixing the reference point. This shows that the proposed method, which limits the number of sampling points and affords sampling position freedom, reduces the computational amount and is effective. When using fixed reference points, the performance increases as the number of points increases; however, this process is inefficient because it requires considerably more points than the deformable method does.

**Effect of Boundary Decoder and BFM.** In Table 2, (e), (f), and (g) present the effects of the proposed boundary decoder and BFM. If only the boundary decoder is used without BFM, the generated boundary map is concatenated with the decoder features without BFM. The results show that, using the boundary map generated by the boundary decoder as a guide, boundary context information is extracted and effectively fused by BFM. This shows that boundary prediction map generation aids in the model’s ability to distinguish between the delicate foreground and background boundaries of camouflaged objects.

### Table 2. Performance with different combinations of our contributions on the CAMO [21], COD10K [10], and CHAMELEON [32] datasets.

| Index | Component | Baseline | MFFM | GLA Transformer | Boundary Decoder | BFM | COD10K [10] | CHAMELEON [32] |
|-------|-----------|----------|------|-----------------|------------------|----|-------------|-----------------|
| (a)   |           | ✓        |      |                 |                  |    | 0.791 0.799 | 0.806 0.811    |
| (b)   |           | ✓        | ✓    |                 |                  |    | 0.793 0.803 | 0.802 0.816    |
| (c)   |           | ✓        | ✓    |                 |                  |    | 0.799 0.800 | 0.806 0.817    |
| (d)   |           | ✓        | ✓    |                 |                  | ✓  | 0.799 0.800 | 0.806 0.817    |
| (e)   |           | ✓        | ✓    |                 |                  | ✓  | 0.803 0.807 | 0.810 0.816    |
| (f)   |           | ✓        | ✓    |                 |                  | ✓  | 0.804 0.808 | 0.816 0.817    |
| (g)   |           | ✓        | ✓    |                 |                  | ✓  | 0.811 0.811 | 0.817 0.817    |
5. Conclusion

In this paper, we propose a GLaDOS model for COD that integrates global information and background and object boundary local information. The proposed model obtains global information from channel-separated features and effectively extracts important local information from the patches using the deformable point sampling method. Furthermore, our method achieves state-of-the-art performance on three popular datasets and various ablation studies show the effectiveness of the proposed model.

References

[1] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Susstrunk. Frequency-tuned salient region detection. In 2009 IEEE conference on computer vision and pattern recognition, pages 1597–1604. IEEE, 2009.

[2] Nagappa U Bhajantri and P Nagabhushan. Camouflage detection identification: a novel approach. In 9th International Conference on Information Technology (ICIT’06), pages 145–148. IEEE, 2006.

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence, 40(4):834–848, 2017.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009.

[5] Bo Dong, Mingchen Zhuge, Yongxiong Wang, Hongbo Bi, and Geng Chen. Towards accurate camouflaged object detection with mixture convolutional nets, atrous convolution, and fully connected crfs. arXiv preprint arXiv:2101.05687, 2021.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020.

[7] Deng-Ping Fan, Ming-Ming Cheng, Jiang-Jiang Liu, Shang-Hua Gao, Qibin Hou, and Ali Borji. Salient objects in clutter: Bringing salient object detection to the foreground. In Proceedings of the European conference on computer vision (ECCV), pages 186–202, 2018.

[8] Deng-Ping Fan, Ming-Ming Cheng, Yun Liu, Tao Li, and Ali Borji. Structure-measure: A new way to evaluate foreground maps. In Proceedings of the IEEE international conference on computer vision, pages 4548–4557, 2017.

[9] Deng-Ping Fan, Cheng Gong, Yang Cao, Bo Ren, Ming-Ming Cheng, and Ali Borji. Enhanced-alignment measure for binary foreground map evaluation. arXiv preprint arXiv:1805.10421, 2018.

[10] Deng-Ping Fan, Ge-Peng Ji, Guolei Sun, Ming-Ming Cheng, Jianbing Shen, and Ling Shao. Camouflaged object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 2777–2787, 2020.

[11] Xue Feng, Cui Guoying, and Song Wei. Camouflage texture evaluation using saliency map. In Proceedings of the Fifth International Conference on Internet Multimedia Computing and Service, pages 93–96, 2013.

[12] Jun Fu, Jing Liu, Hajie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hangqing Lu. Dual attention network for scene segmentation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 3146–3154, 2019.

[13] Shang-Hua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip Torr. Res2net: A new multi-scale backbone architecture. IEEE TPAMI, 2020.

[14] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415, 2016.

[15] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 11108–11117, 2020.

[16] Iván Huerta, Daniel Rowe, Mikhail Mozerev, and Jordi González. Improving background subtraction based on a casuistry of colour-motion segmentation problems. In Iberian Conference on Pattern Recognition and Image Analysis, pages 475–482. Springer, 2007.

[17] Ge-Peng Ji, Lei Zhu, Mingchen Zhuge, and Keren Fu. Fast camouflaged object detection via edge-based reversible recalibration network. Pattern Recognition, 123:108414, 2022.

[18] Qi Jia, Shutian Yao, Yu Liu, Xin Fan, Risheng Liu, and Zhongxuan Luo. Segment, magnify and reiterate: Detecting camouflaged objects the hard way. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4713–4722, 2022.

[19] Ch Kavitha, B Prabhakara Rao, and A Govardhan. An efficient content based image retrieval using color and texture of image sub block. International Journal of Engineering Science and Technology (IJEST), 3(2):1060–1068, 2011.

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[21] Trung-Nghia Le, Tam V Nguyen, Zhongliang Nie, Minh-Triet Tran, and Akihiro Sugimoto. Anabranch network for camouflaged object segmentation. Computer Vision and Image Understanding, 184:45–56, 2019.

[22] Minhyeok Lee, Chaewon Park, Suwhan Cho, and Sangyoun Lee. Spsn: Superpixel prototype sampling network for rgb-d salient object detection. arXiv preprint arXiv:2207.07898, 2022.

[23] Aixuan Li, Jing Zhang, Yunqiu Lv, Bowen Liu, Tong Zhang, and Yuchao Dai. Uncertainty-aware joint salient object and camouflaged object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10071–10081, 2021.
[24] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2171–2180, 2017. 3

[25] Jiwei Liu, Jing Zhang, and Nick Barnes. Modeling aleatoric uncertainty for camouflaged object detection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 1445–1454, 2022. 7

[26] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983, 2016. 7

[27] Yunqiu Lv, Jing Zhang, Yuchao Dai, Aixuan Li, Bowen Liu, Nick Barnes, and Deng-Ping Fan. Simultaneously localize, segment and rank the camouflaged objects. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 11591–11601, 2021. 6, 7

[28] Haiyang Mei, Ge-Peng Ji, Ziqi Wei, Xin Yang, Xiaopeng Wei, and Deng-Ping Fan. Camouflaged object segmentation with distraction mining. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8772–8781, 2021. 6, 7

[29] Chaewon Park, Minhyeok Lee, MyeongAh Cho, and Sangyoun Lee. Saliency detection via global context enhanced feature fusion and edge weighted loss. arXiv preprint arXiv:2110.06550, 2021. 2

[30] Suo Qiu. Global weighted average pooling bridges pixel-level localization and image-level classification. arXiv preprint arXiv:1809.08264, 2018. 4

[31] Jingjing Ren, Xiaowei Hu, Lei Zhu, Xuemiao Xu, Yangyang Xu, Weiming Wang, Zijun Deng, and Pheng-Ann Heng. Deep texture-aware features for camouflaged object detection. IEEE Transactions on Circuits and Systems for Video Technology, pages 1–1, 2021. 1, 2, 7

[32] Przemysław Skurowski, Hassan Abdulameer, J Blaszczyk, Tomasz Depta, Adam Kornacki, and P Kozieł. Animal camouflage analysis: Chameleon database. Unpublished manuscript, 2(6):7, 2018. 2, 6, 7, 8

[33] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 7262–7272, 2021. 4

[34] Yujia Sun, Geng Chen, Tao Zhou, Yi Zhang, and Nian Liu. Context-aware cross-level fusion network for camouflaged object detection. arXiv preprint arXiv:2105.12555, 2021. 1, 2, 7

[35] Yujia Sun, Shuo Wang, Chenglizhao Chen, and Tian-Zhu Xia. Boundary-guided camouflaged object detection. arXiv preprint arXiv:2207.00794, 2022. 1, 2, 6, 7

[36] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7794–7803, 2018. 5

[37] Jun Wei, Shuhui Wang, and Qingming Huang. F^3 net: fusion, feedback and focus for salient object detection. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 12321–12328, 2020. 5

[38] Tiange Xiang, Chaoyi Zhang, Yang Song, Jianhui Yu, and Weidong Cai. Walk in the cloud: Learning curves for point clouds shape analysis. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 915–924, 2021. 5

[39] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. Advances in Neural Information Processing Systems, 34:12077–12090, 2021. 4

[40] Fan Yang, Qiang Zhai, Xin Li, Rui Huang, Ao Luo, Hong Cheng, and Deng-Ping Fan. Uncertainty-guided transformer reasoning for camouflaged object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 4146–4155, 2021. 7

[41] Yuhui Yuan, Xiulin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In European conference on computer vision, pages 173–190. Springer, 2020. 3

[42] Qiang Zhai, Xin Li, Fan Yang, Chenglizhao Chen, Hong Cheng, and Deng-Ping Fan. Mutual graph learning for camouflaged object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 12997–13007, 2021. 2, 7

[43] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In International conference on machine learning, pages 7354–7363. PMLR, 2019. 5

[44] 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 Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 8779–8788, 2019. 2

[45] Sixiao Zheng, Jiachen Lu, Xian Zou, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In CVPR, 2021. 4

[46] Yijie Zhong, Bo Li, Lv Tang, Senyun Kuang, Shuang Wu, and Shouhong Ding. Detecting camouflaged object in frequency domain. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4504–4513, 2022. 2

[47] Hongwei Zhu, Peng Li, Haoran Xie, Xuefeng Yan, Dong Liang, Dapeng Chen, Mingqiang Wei, and Jinfeng Qi. I can find you! boundary-guided separated attention network for camouflaged object detection. AAAI, 2022. 1, 2, 6, 7

[48] Jinchao Zhu, Xiaoyu Zhang, Shuo Zhang, and Junnan Liu. Boundary-guided separated attention network for camouflaged object detection. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pages 3599–3607, 2021. 1, 2, 7

[49] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159, 2020. 4