Predictive Uncertainty Estimation for Camouflaged Object Detection

Yi Zhang, Jing Zhang, Wassim Hamidouche, Olivier Deforges

Predictive Uncertainty Estimation for Camouflaged Object Detection

Yi Zhang, Jing Zhang, Wassim Hamidouche, Olivier Deforges

***Abstract*—Uncertainty is inherent in machine learning methods, especially those for camouflaged object detection aiming to finely segment the objects concealed in background. The strong "center bias" of the training dataset leads to models of poor generalization ability as the models learn to find camouflaged objects around image center, which we define as "model bias". Further, due to the similar appearance of camouflaged object and its surroundings, it is difficult to label the accurate scope of the camouflaged object, especially along object boundaries, which we term as "data bias". To effectively model the two types of biases, we resort to uncertainty estimation and introduce predictive uncertainty estimation technique, which is the sum of model uncertainty and data uncertainty, to estimate the two types of biases simultaneously. Specifically, we present a predictive uncertainty estimation network (*PUENet*) that consists of a Bayesian conditional variational auto-encoder (BCVAE) to achieve predictive uncertainty estimation, and a predictive uncertainty approximation (PUA) module to avoid the expensive sampling process at test-time. Experimental results show that our *PUENet* achieves both highly accurate prediction, and reliable uncertainty estimation representing the biases within both model parameters and the datasets.**

***Index Terms*—Uncertainty estimation, segmentation, camouflaged objects.**

I. Introduction

DURING the past few years, researches such as [1]–[13] apply machine learning to address camouflaged object detention (COD), which aims to finely segment the objects concealed in realistic natural environment (examples are shown in Fig. 1). Though significant improvement has been achieved, current COD methods fail to explore deeply towards the interpretability of their methodologies and intrinsic bias existed in training datasets.

The objective of machine learning methods is to minimize the empirical loss function [15] as:

$$\begin{aligned}\min_{\theta} \mathbb{E}_{X,Y}[\mathcal{L}(f(X;\theta),Y)] &= \int \mathcal{L}(f(X;\theta),Y)dp(X,Y) \\ &\approx \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(f(x_i;\theta),y_i), \quad (x_i,y_i)\sim p(X,Y),\end{aligned} \tag{1}$$

where θ is the learned parameter set of model $f(\cdot)$ and (x_i, y_i) denotes sampled pair from the joint data distribution $p(X, Y)$. X, Y are input and output variables from the training dataset D. $\mathcal{L}(\cdot)$ is the loss function. Given D, the unawareness of the task and the data leads to at least two types of biases: 1) model bias representing our ignorance of the model and 2) data bias, indicating the inherent noise within the dataset. The former can be addressed with enough diverse training datasets to provide enough knowledge about the task, while the latter is usually caused by inherent noise, *i.e.*, precision of the sensors, accuracy of labeling, which is difficult to be completely avoided. COD suffers greatly from above two types of biases. Camouflaged objects are objects that hide in the environment, which usually share similar appearance as their surroundings. In realistic scenes, there should not exist any location-related prior for camouflaged objects. However, as the existing COD training datasets (*e.g.*, [1], [3]) are collected from the Internet where the photographers focus on the camouflaged instance, causing serious "center bias" issue where most of the camouflaged instances locate at or near the center of the image. The COD model trained with the center-biased training dataset may fail to generalize well as it greatly reduces the searching space for COD, thus introducing "model bias". Further, unlike other object segmentation tasks where clear boundaries exist between targets and their surroundings, camouflaged objects share similar appearance as the environment, making it hard for annotators to precisely locate and depict camouflaged objects, leading to inherent noise within the manual labels.

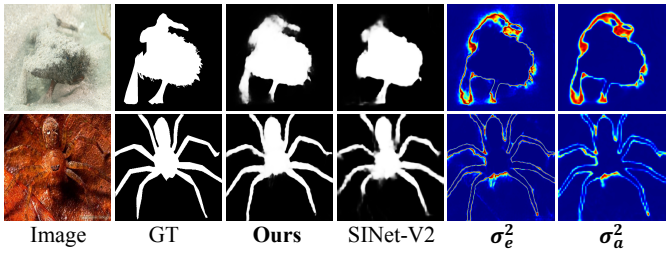

Fig. 1. An example illustrating COD and uncertainty estimation. "σ_e^2" is the sampling-based uncertainty of "Bayesian conditional variational auto-encoder" (BCVAE). "σ_a^2" is the output of "predictive uncertainty approximation" (PUA) module. SINet-V2 [14] is a state-of-the-art method.

To model the above two types of biases, we resort to uncertainty estimation [16], which is a mechanism to understand model limitations. In this paper, we use uncertainty to explain the two types of biases for COD. Particularly, [16] define model bias as epistemic uncertainty, and data bias as aleatoric uncertainty estimation. We argue that both types of uncertainty exist in COD, and we define them together as predictive uncertainty following [17]. Most of the existing techniques for predictive uncertainty estimation are based on Bayesian

Yi Zhang, Wassim Hamidouche and Olivier Deforges are with Univ Rennes, INSA Rennes, CNRS, IETR (UMR 6164), France.

Jing Zhang is with School of Computing, Australian National University, Canberra, Australia.

The source code and experimental results are publicly available via our project page: https://github.com/Jun-Pu/PUENet.

neural network (BNN), where a pre-defined prior distribution $p(\theta)$ is set as a constraint to regularize the distribution of the model parameters, leading to stochastic predictions. At test-time, multiple iterations of samplings are performed, and the mean prediction and uncertainty (entropy of the mean value of multiple predictions) are obtained for deterministic model evaluation and prediction confidence estimation.

We find that the BNN based uncertainty estimation methods are less efficient when applied in real life as expensive sampling process is needed to achieve effective uncertainty estimation. In this work, we introduce a sampling-free **p**redictive **u**ncertainty **e**stimation **net**work (*PUENet*) for COD via a Bayesian latent variable model. With the extra latent variable as origin of predictive uncertainty [18], we aim to achieve both accurate predictions and reliable uncertainty maps (Fig. 1), leading to explainable camouflage modeling. In addition, considering the ambiguity between camouflaged object and its background, we propose "selective attention module" (SAM) which automatically selects and refines the hierarchical features by combining the high-level-based prediction and its reversal with channel-attention mechanism [19].

We summarize our main contributions as follows: 1) We introduce *PUENet* to explicitly model both model uncertainty and data uncertainty for effective COD. 2) We demonstrate that both uncertainties are important for COD, and simultaneously modeling them results in better performance and reliable uncertainty maps explaining the limitations of the trained model. 3) We propose SAM to automatically refine the hierarchical features, leading to more effective feature representation for camouflaged objects with complex background.

II. Related Work

In this section, we introduce related works from the aspects of COD, model/data bias, uncertainty estimation and attention mechanisms. We also highlight the uniqueness of our *PUENet*, compared to existing COD methods.

Camouflaged Object Detection (COD): [20] is a pioneer work that conducted COD in videos by modeling motion cues with heuristic methods, while recently proposed method [2] addressed video COD with deep learning methods. As the establishment of large-scale image COD datasets [1], [3], utilizing deep neural networks (DNNs) to detect concealed objects in static images has become the mainstream in the field. ANet [1] brought (non-)/camouflage classification as an auxiliary task for the COD. Taking advantage of the multi-task architecture, LSR [4] learned to simultaneously localize, rank and segment the concealed objects. SINet [3] designed a two-stage end-to-end architecture to respectively mimic the searching and identifying procedures of hunting in the wild. With the same motivation, D^2C-Net [5] obtained better performance on the same benchmark. From the cognitive perspective, MirrorNet [6] applied bio-inspired attack stream to aid the COD. PreyNet [21] added a bi-directional attention module and a pixel-wise label-based calibration module to the vanilla U-Net like architecture, to mimic the predator-prey interaction in a progressive manner. Moreover, TINet [7] and BASNet [8] took a deeper look of texture and boundary cues of camouflaged objects. Continuously focusing on the texture information, TANet [9] designed a texture extractor to refine the encoded features. Based on similar observation of objects' texture, SINet-v2 [14] further improved the original structure by adding a texture enhanced module. In addition, C^2FNet [10], MCIF-Net [11], PFNet [12] and MGL [13] paid attention to the global context awareness by proposing multi-scale attention strategies which automatically select and fuse the useful multi-level features for effective decoding. Most recent work such as ZoomNet [22] gained improved performance via mimicking the "zoom strategy" of human vision system. SegMaR [23] proposed an iterative refinement framework to segment camouflaged targets at multi-resolution. HitNet [24] used high-resolution-based features to iteratively refine the low-resolution-based ones. DGNet [25] used image-based gradients to support the refinement of texture-based features. FSPNet [26] applied non-local attention mechanism to process high-level features of the vision transformer-based backbone, to advance COD performance. Besides merely taking advantage of RGB information, FDCOD [27] and FEDER [28] explored visual representation of camouflaged objects in the frequency domain via discrete Cosine transform and wavelet transform techniques, respectively. CRNet [29] proposed new consistency loss to enable the training of COD model with only scribble annotations. Besides binary segmentation, CFL [30] conducted instance-level COD on newly proposed CAMO++ dataset. More statistics and discussions towards recent COD datasets and methodologies are detailed in [31].

Model Bias: One of the most commonly seen bias in current COD datasets is the center bias (or model bias), due to a photographers' tendency of framing the concealed objects centered on the images. The issue also often occurs in saliency detection datasets and has been widely discussed in both fixation prediction and salient object detection [32]. As a result, such a center-biased human visual attention pattern can be explicitly or implicitly modeled to aid automatic saliency judgements. Early work such as [33] introduced a center prior indicating the distance to the center of each of the pixels. [34] added a location biased convolutional layer to DNNs to learn the location-based pattern of specific datasets. Later work [35] applied VGG architecture [36] with proposed loss function to predict saliency map formulated as a generalized Bernoulli distribution. More recently, [37] concatenated a learned Gaussian map at the bottleneck to inject the center bias prior to the high level features. "Center bias" for saliency detection is useful prior as it is consistent with human perception. However, it greatly reduces the searching space for COD, and model trained with center-biased data may be less effective to localize camouflaged objects elsewhere.

Data Bias: In this paper, we refer "data bias" as difficulty in labeling, leading to noisy labeling. Learning from noisy labels, which can be introduced by idiosyncrasies and errors of annotators, has been widely studied in recent years. We discuss two main directions. *Architecture*, *i.e.* dropout noise model [38], which learned the dataset-specific label transition process by adding a noise adaption layer at the bottleneck of DNN architecture. Another type of methods (*e.g.* [39]) tackle multi-

source noisy labeling via specifically designed structures. *Regularization* [40]–[44] is another research direction for learning from noisy label. Generally, the model regularization can be conducted either explicitly (*e.g.*, robust early-learning [45]) or implicitly (*e.g.*, adversarial training [46]).

Uncertainty Estimation: Uncertainty represents model ignorance about its prediction. [16] defined two types of uncertainty, namely epistemic uncertainty and aleatoric uncertainty. The former is caused by model's limited knowledge, and the latter is usually due to inherent noise within the data. BNN is a straightforward solution for epistemic uncertainty modeling by putting a distribution over model parameters. However, it is intractable to perform the posterior inference within BNN due to the intractable integral operation. Then, variational inference was proposed as approximation of posterior inference. Among them, Monte-Carlo dropout (MC-Dropout) [47] was widely studied, which approximates the intractable integral with MC integration. Deep ensembles [48] is another commonly applied model uncertainty estimation method, where multiple models with the same base model (*e.g.*, backbone) were trained. The mean of models' outputs were then used to compute the uncertainty. To model the aleatoric uncertainty, [16] suggested training an extra uncertainty estimation branch, which serves as both the weight and the regularizer of the original task related loss function.

Attention Models: Squeeze-excitation (SE) attention [19] emphasized the channel-wise effective features by squeezing the spatial features with an adaptive average pooling layer and by computing channel-wise attention using two fully-connected (FC) layers. [49] further proposed a three-stage (*i.e.*, splitting, fusion and selection) attention mechanism, where the input features are split into multiple branches and convolved with different kernels. The processed features are then fused with SE attentions and summed as final output. Also based on SE mechanism, ECA [50] focused on computing local adjacent channel attention by replacing the two FC layers of SE model with an 1D-convolutional layer. Besides above channel attention-based models, [51] used a large kernel (*e.g.*, 7×7) to further extract spatial attention based on channel-wise-refined features. Similarly, [52] also applied both the channel and spatial attentions to feature refinement. However, it simply sums the attention matrix, rather than cascading the channel-/spatial-based ones as in [51]. In addition, self-attention [53] are widely used in the fields of not only computer vision but also natural language processing and multi-modal learning. Self-attention is a type of operation where the input feature is first mapped to "query", "key" and "value" features via FC layers, respectively. The final output feature is computed as the output of a dot product of "value" and the result of a dot product of "query" and "key". [54] further combined the ideas of channel-spatial-attention and self-attention mechanisms and thus proposing a duel-attention model for scene segmentation. Co-attention networks such as [55], [56], also inspired by self-attention, were proposed to learn complementary information between different visual cues.

Uniqueness of Our Solution: Being different from existing COD models (*e.g.*, [12], [14]), we produce uncertainty along with the predictions, leading to explainable COD. Existing uncertainty-aware COD models, *e.g.*, [57], [58], either uses uncertainty for hard-negative mining [57] at test-time where an expensive sampling process is performed, or resorts [58] to Generative Adversarial Net [59] for stochastic predictions. Our "single-pass" predictive uncertainty estimation method is efficient to use at both train and test times. Further, although conditional variational auto-encoder (CVAE) [60] has been used in [61] and it's extension [62] for the "subjective nature" of saliency modeling via multiple iterations of sampling, our Bayesian latent variable model explore CVAE for predictive uncertainty estimation. Most recent work such as UDASOD [63] directly used variance maps of multiple predictions based on data augmentation such as scale and flip, to facilitate the pseudo-label generation for the task of unsupervised domain adaptive salient object detection (SOD). Our predictive uncertainty approximation (PUA) solution avoids multiple samplings and does not rely on any types of data augmentation. UMNet [64] resorts to the uncertainty maps of multiple hand-crafted SOD models, to enable the training of its unsupervised segmentation head. Our network learns to segment objects without any domain priors brought by external models. Also, our SAM module is specifically designed for COD to automatically identify and refine the hard region predictions, resulting in more accurate predictions.

III. Methodology

In this section, we first introduce the uncertainty estimation techniques. Then, we discuss the proposed sampling-free predictive uncertainty estimation method (*PUENet*) for COD. The pipeline of our framework is shown in Fig. 2.

A. Uncertainty Estimation Techniques

There mainly exists two types of uncertainty, namely epistemic uncertainty explaining the model bias and aleatoric uncertainty for data bias estimation [16]. The former can be explained away with large amount and diverse training datasets [16], while the latter is caused by inherent noise, which usually cannot be explained away.

Epistemic Uncertainty Estimation: To model epistemic uncertainty, one can change the deterministic neural network into stochastic neural network by putting a prior distribution over network parameters, *e.g.*, $p(\theta)$. In this way, a BNN is achieved. Compared with deterministic neural network that optimizes model parameters directly, BNN inference is achieved through marginalisation, where multiple predictions with respect to all possible model parameters are averaged for model optimization. [68] then defines the epistemic uncertainty within the BNN as the mutual information of model prediction and model parameters.

Aleatoric Uncertainty Estimation: The aleatoric uncertainty captures the observation noise δ inherent in the training dataset D. To perform variational inference over the noise level δ, non-BNN is used to obtain the deterministic model parameter set θ. To obtain aleatoric uncertainty, maximum likelihood estimation is adopted with the additive labeling noise assumption. Thus, the final aleatoric uncertainty (conditioning on the input image) is usually modeled with an

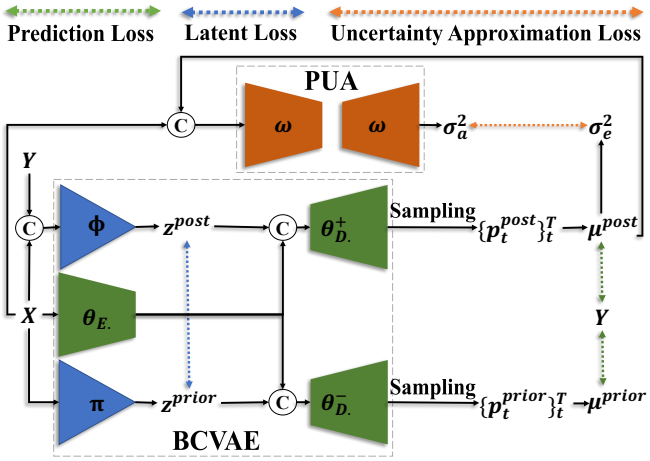

Fig. 2. The training pipeline of our *PUENet*, which consists of a "Bayesian conditional variational auto-encoder" (BCVAE), and a "predictive uncertainty approximation" (PUA) module. In BCVAE, the prior/posterior-based decoders share an identical architecture yet with separate parameter sets, *i.e.*, $\theta_{D.}^{-}/\theta_{D.}^{+}$. Similarly, the prior/posterior distribution estimation modules have the same model structure however with different parameter sets (π/ϕ). "σ_e^2" and "σ_a^2" denote the sampling-based uncertainty and approximated uncertainty, respectively. Detailed structures of BCVAE and PUA are shown in Fig. 3.

extra uncertainty estimation module, where its output serves as both the weight and the regularization of the original task-related loss function [16]. In this way, the aleatoric uncertainty is constrained with the model's loss function that pushes it to a specific range. Specifically, the aleatoric uncertainty can be defined as the mean entropy of multiple predictions: $\frac{1}{T}\sum_{t=1}^{T}\mathbb{H}[p(Y|X;\theta_t,\phi)]$, where $\mathbb{H}[\cdot]$ is the entropy operation.

Predictive Uncertainty Estimation: [17] defines "predictive uncertainty" as the sum of the aleatoric uncertainty and epistemic uncertainty. The typical way to achieve predictive uncertainty is through a BNN with input-level or feature-level noise injection. During test-time, the predictive uncertainty is then defined as entropy of the mean prediction.

B. Predictive Uncertainty Estimation Network

One main issue with above two types of uncertainty modeling is that they rely on multiple times of sampling during test-time to achieve uncertainty estimation, which is less efficient when used in real life. In this paper, we introduce predictive uncertainty estimation to achieve sampling-free uncertainty estimation. Specifically, we design a BNN to capture the distribution of model parameters. Further, we add extra inference model and adapt our network to a conditional variational auto-encoder (CVAE) [60], which is used to model the distribution of model prediction. In this way, our framework can estimate both model uncertainty (with the BNN) and the data uncertainty (with the CVAE). Further, we present predictive uncertainty approximation (PUA) module to approximate the sampling-based predictive uncertainty of the proposed Bayesian conditional variational auto-encoder.

BCVAE Framework: For a BNN, a pre-defined prior distribution $p(\theta)$ is set as a constrain to regularize the distribution of the model parameters θ, thus it can achieve stochastic predictions. According to Bayes' rule, the posterior over model parameters $p(\theta|X,Y)$ can be achieved by $p(\theta|X,Y) = p(Y|X,\theta)p(\theta)/p(Y|X)$. As the marginal likelihood $p(Y|X)$ cannot be evaluated analytically, the posterior of model parameter $p(\theta|X,Y)$ is then difficult to be evaluated as well. As an approximation, some inference techniques approximate the real posterior $p(\theta|X,Y)$ with a simple distribution $q(\theta|\gamma)$, where γ is used to control the distribution of θ. In this way, the intractable Bayesian inference is replaced with a simple optimization that optimizes over the hyper-parameter γ.

In this paper, we adopt the Monte Carlo integration [47] as an approximation of the intractable $p(Y|X)$:

$$p(Y|X) \approx \frac{1}{T}\sum_{t=1}^{T} p(Y|X,\theta_t), \tag{2}$$

where $\theta_t \sim q(\theta|\lambda)$, T is the times of sampling. When we define stochastic of model parameter set θ as randomly multiplying the output of each neuron by a binary mask drawn from a Bernoulli distribution, we achieve MC dropout [47], where λ is the Bernoulli distribution related hyper-parameter. To achieve this, following [69], we add dropout after each level features of the backbone network. With the MC dropout based BNN, we can effectively estimate model uncertainty by modeling distribution of model parameter set θ, or what we claim as the "model bias".

To model the "data bias", we further introduce CVAE [60] to estimate the distribution of model prediction $p(Y|X;\theta)$. Specifically, following the conventional practice of CVAE, we introduce two extra encoders to our BCVAE network to model both the prior and posterior distribution of the latent variable, namely the prior distribution estimation model $p_\pi(z|X)$ and posterior distribution estimation model $p_\phi(z|X,Y)$, where π and ϕ are network parameter sets of the two sets of encoders, z is the latent variable modeling the "data bias".

Learning a CVAE framework involves approximation of the true posterior distribution of z with the designed inference model $p_\phi(z|X,Y)$. The parameter sets of CVAE can be estimated in stochastic variational Bayes [70] framework by maximizing the evidence lower bound (ELBO) as:

$$\begin{aligned} L(\theta,\phi,\pi;X) = \mathbb{E}_{z\sim p_\phi(z|X,Y)}[\log(p_\theta(Y|X,z))] \\ - D_{KL}(p_\phi(z|X,Y)||p_\pi(z|X)), \end{aligned} \tag{3}$$

where $D_{KL}(p_\phi(z|X,Y)||p_\pi(z|X))$ penalizes the divergence between the posterior and prior distribution of z.

With the BCVAE framework, we achieve both model parameter set distribution estimation and prediction distribution estimation, making it possible to estimate predictive uncertainty, which is the total of model uncertainty (the BNN part) and data uncertainty (the CVAE part). Specifically, we define the mean model prediction as:

$$\mu(Y|X) = \frac{1}{T}\sum_{t=1}^{T} p(Y|X;\theta_t,\phi), \tag{4}$$

and then the predictive uncertainty is defined as entropy of the mean prediction $\sigma_p^2 = \mathbb{H}[\mu(Y|X)]$.

As shown in Eq. 4, we need to sample multiple times to achieve the mean model prediction as well as the predictive uncertainty estimation, which is less efficient when used in

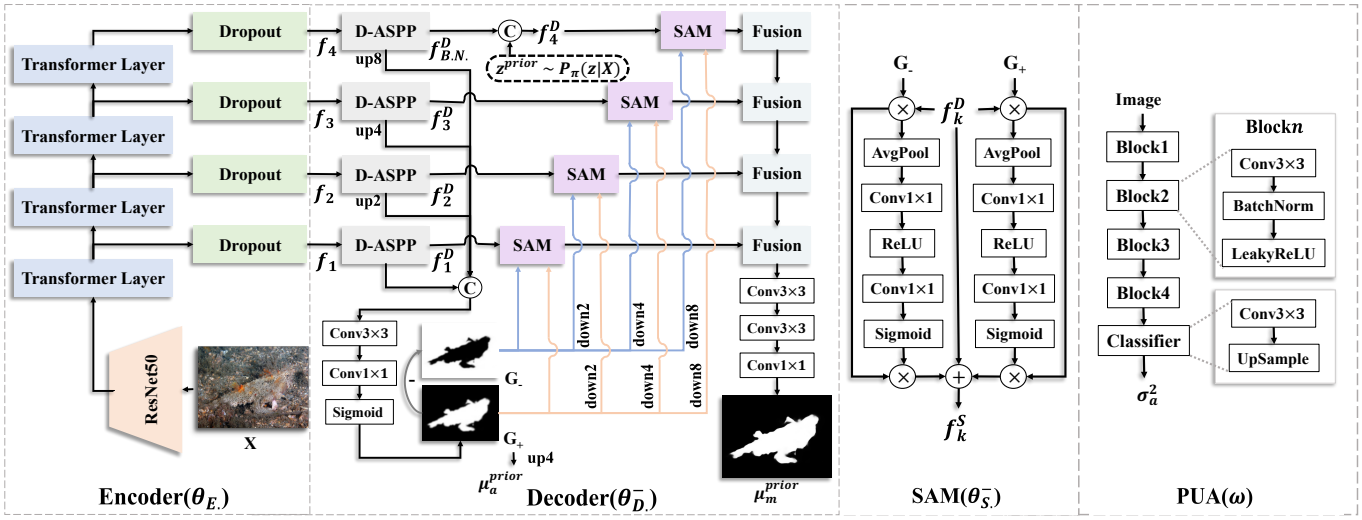

Fig. 3. Architectures of "predictive uncertainty approximation" (PUA) module (ω), and "Bayesian conditional variational auto-encoder" BCVAE's encoder ($\theta_{E.}$)/prior-based decoder($\theta_{D.}^{-}$). Note that we use hybrid-ViT-based backbone [65] as an example, and by replacing it with pure ResNets, we obtain the other two models as shown in Table I. SAM denotes "selective attention module" (note that $\theta_{S.}^{-} \subset \theta_{D.}^{-}$). D-ASPP means the "DenseASPP block" [66]. Fusion is "residual convolutional fusion block" from MiDaS [67]. "z^{prior}" denotes latent variable from prior distribution estimation module (π), as illustrated in Fig. 2).

real life. We further introduce PUA to achieve sampling-free uncertainty estimation at test-time.

PUA Framework: Inspired by confidence-aware learning from [71], we intend to approximate the sampling based predictive uncertainty with a designed uncertainty estimation module. Specifically, we introduce extra PUA module $f_\omega(X, p(Y|X;\theta,\phi))$ to estimate the inherent uncertainty given the input images X and current prediction $p(Y|X;\theta,\phi)$, where ω is the parameter set of the PUA module. To train PUA, we define cross entropy loss between $f_\omega(X, p(Y|X;\theta,\phi))$ and the sampling based uncertainty σ_e^2: $\mathcal{L}(f_\omega(X, p(Y|X;\theta,\phi)), \sigma_e^2)$. During test-time, we use the proposed PUA module to generate the approximated predictive uncertainty (σ_a^2) to avoid the expensive sampling process.

We further illustrate the structural details of the BCVAE, which consists of BNN and prior/posterior distribution estimation modules (for CVAE), and PUA.

BNN Architecture: Our BNN for COD is an end-to-end encoder-decoder network (Fig. 3), with flexible choices of backbones such as ResNet50 [72], Res2Net50 [73] and hybrid-ViT [74]. The decoder consists of three main components, *i.e.*, hierarchical DenseASPP [66] blocks, residual convolutional fusion layers [67], and the proposed selective attention modules (SAM). Specifically, our SAM consists of two channel attention [19] based modules which select and refine the guidance ($\{G_+, G_-\}$) weighted features (Fig. 3), where G_+ and G_- are high-level features-based prediction and its reversal, respectively. With $\{G_+, G_-\}$ and backbone features after the kth DenseASPP [66] block ($f_{\mathrm{k}}^{\mathrm{D}}$), the refined features $f_{\mathrm{k}}^{\mathrm{S}}$ can be computed as:

$$\begin{aligned} f_{\mathrm{k}}^{\mathrm{S}} = CA(G_+ \otimes f_{\mathrm{k}}^{\mathrm{D}}) \otimes (G_+ \otimes f_{\mathrm{k}}^{\mathrm{D}}) + \\ CA(G_- \otimes f_{\mathrm{k}}^{\mathrm{D}}) \otimes (G_- \otimes f_{\mathrm{k}}^{\mathrm{D}}) + f_{\mathrm{k}}^{\mathrm{D}}, \end{aligned} \tag{5}$$

where $CA(\cdot) = \sigma(Conv(ReLU(Conv(P(\cdot)))))$ is the channel attention module. σ means Sigmoid function, $P(\cdot)$ is average pooling layer. $\otimes$ represents Hadamard product.

Prior/Posterior Distribution Estimation Modules: We borrow the architecture of latent feature encoder model in [61] to estimate the prior and posterior distribution of the latent variable z respectively. Specifically, the prior distribution estimation model takes images X as input, and output $\{\mu^{prior}, \sigma^{prior}\}$, where μ^{prior} and σ^{prior} are mean and standard deviation of the prior distribution. The posterior distribution model takes the concatenation of images X and ground truth Y as input to model the structured latent space distribution with mean and standard deviation pair $\{\mu^{post}, \sigma^{post}\}$. The latent variable of both distributions is then obtained with the re-parameterization trick as: $z^{prior} = \mu^{prior} + \sigma^{prior} \odot \epsilon$, and $z^{post} = \mu^{post} + \sigma^{post} \odot \epsilon$, where $\epsilon \sim \mathcal{N}(0,1)$, and $\odot$ is the dot-product operation.

Algorithm 1 Training PUENet

Input: (1) Training images $\{x_i\}_i^n$ with associated ground truth (GT) $\{y_i\}_i^n$; (2) Maximum of learning iterations M.
Output: Parameters θ for the BNN, π and ϕ for the prior and posterior distribution estimation modules respectively, ω for the PUA module (details are shown in Fig. 2).

1: Initialize θ, π, ϕ and ω
2: **for** $t \leftarrow 1$ to M **do**
3: Sample image-GT pairs $\{(x_i, y_i)\}_i^b$, b is batch size.
4: For each x_i, sample the prior $z_i^{prior} \sim p_\pi(z|x_i)$ for T times. Compute the prior-based BCVAE mean prediction μ_i^{prior}.
5: For each (x_i, y_i), sample the posterior $z_i^{post} \sim p_\phi(z|x_i, y_i)$ for T times, and compute the posterior-based BCVAE mean prediction μ_i^{post} and uncertainty σ_{ei}^2.
6: For each (x_i, μ_i^{post}), compute the uncertainty σ_{ai}^2 via PUA.
7: Update the parameters of BCVAE (θ, π and ϕ) and PUA (ω) together via loss Eq. 6.
8: **end for**

TABLE I
PERFORMANCE COMPARISON WITH STATE-OF-THE-ART COD MODELS ON BENCHMARK TESTING DATASETS.↑ INDICATES THE HIGHER THE SCORE THE BETTER, AND VICE VERSA FOR ↓. "TR.SIZE" DENOTES THE INPUT IMAGE SIZE FOR THE MODEL TRAINING. * DENOTES MODELS TRAINED ON MULTI-SCALE. THE BEST RESULT OF EACH COLUMN IS IN **RED**. F_β, E_ξ, S_α AND $\mathcal{M}$ INDICATE MEAN F-MEASURE, MEAN E-MEASURE, S-MEASURE ($\alpha = 0.5$) AND MEAN ABSOLUTE ERROR, RESPECTIVELY. ‡ DENOTES NO INFORMATION PROVIDED.

Method	Tr.Size	Backbone	Year	CAMO [1]				CHAMELEON [75]				COD10K [3]				NC4K [4]			
				$S_\alpha\uparrow$	$F_\beta\uparrow$	$E_\xi\uparrow$	$\mathcal{M}\downarrow$	$S_\alpha\uparrow$	$F_\beta\uparrow$	$E_\xi\uparrow$	$\mathcal{M}\downarrow$	$S_\alpha\uparrow$	$F_\beta\uparrow$	$E_\xi\uparrow$	$\mathcal{M}\downarrow$	$S_\alpha\uparrow$	$F_\beta\uparrow$	$E_\xi\uparrow$	$\mathcal{M}\downarrow$
SINet [3]	352^2	ResNet50	CVPR'20	.745	.702	.804	.092	.872	.827	.936	.034	.776	.679	.864	.043	.810	.772	.873	.057
LSR [4]	352^2*	ResNet50	CVPR'21	.793	.725	.826	.085	.893	.839	.938	.033	.793	.685	.868	.041	.839	.779	.883	.053
UJSC [58]	352^2*	ResNet50	CVPR'21	.803	.759	.853	.076	.894	.848	.943	.030	.817	.726	.892	.035	.842	.806	.898	.047
MGL [13]	352^2*	ResNet50	CVPR'21	.775	.726	.812	.088	.893	.834	.918	.030	.814	.711	.852	.035	.833	.782	.867	.052
PFNet [12]	416^2*	ResNet50	CVPR'21	.782	.744	.840	.085	.882	.826	.922	.033	.800	.700	.875	.040	.829	.782	.886	.053
SINet-V2 [14]	352^2*	Res2Net50	TPAMI'21	.820	.782	.882	.070	.888	.835	.942	.030	.815	.718	.887	.037	.847	.805	.903	.048
UGTR [57]	473^2*	ResNet50	ICCV'21	.785	.686	.859	.086	.888	.796	.918	.031	.818	.667	.850	.035	.839	.786	.873	.052
D^2C-Net [5]	320^2	Res2Net50	TIE'21	.774	.735	.818	.087	.889	.848	.939	.030	.807	.720	.876	.037	‡	‡	‡	‡
IEANet [76]	352^2	ResNet50	TCDS'22	.760	‡	.764	.099	.872	‡	.882	.043	.778	‡	.795	.050	‡	‡	‡	‡
ZoomNet [22]	384^2*	ResNet50	CVPR'22	.820	.794	.877	.066	.902	.864	.943	.023	.838	.766	.888	.029	.853	.818	.896	.043
SegMaR [23]	352^2	ResNet50	CVPR'22	.815	.795	.874	.071	.906	.872	.951	.025	.833	.757	.899	.033	.841	.821	.896	.046
FDCOD [27]	416^2	Res2Net50	CVPR'22	.844	‡	.898	.062	.898	‡	.949	.027	.837	‡	.918	.030	‡	‡	‡	‡
BGNet [77]	416^2	Res2Net50	IJCAI'22	.812	.789	.870	.073	.901	.860	.943	.027	.831	.753	.901	.033	.851	.820	.907	.044
BSANet [78]	384^2	Res2Net50	AAAI'22	.796	.763	.851	.079	.895	.858	.946	.027	.818	.738	.891	.034	.841	.808	.897	.048
HitNet [24]	704^2	PVTv2	AAAI'23	.849	.831	.906	.055	.921	.900	.967	.019	.871	.823	.935	.023	.875	.853	.926	.037
DGNet [25]	352^2	EfficientNet	MIR'23	.839	.806	.901	.057	.890	.834	.938	.029	.822	.728	.896	.033	.857	.814	.911	.042
		ResNet50	2023	.794	.762	.857	.080	.888	.844	.943	.030	.813	.727	.887	.035	.836	.798	.892	.050
PUENet	512^2	Res2Net50	2023	.834	.806	.889	.067	.897	.858	.940	.027	.844	.774	.910	.029	.862	.830	.913	.042
(Ours)		Hybrid-ViT	2023	**.877**	**.860**	**.930**	**.045**	.910	.869	.957	.022	**.873**	.812	**.938**	**.022**	**.898**	**.874**	**.945**	**.028**

Algorithm 2 Testing PUENet

Input: Testing images $\{x_i\}_i^n$.
Output: Prediction p_i and predictive uncertainty σ_{ai}^2 of x_i.

```
for i ← 1 to n do
    For each x_i, compute p_i via BCVAE.
    For each (x_i, p_i), compute uncertainty σ²_ai via PUA.
end for
```

PUA Architecture: As shown in Fig. 3, our PUA module consists of four convolutional blocks, aiming to approximate the real sampling based predictive uncertainty with our designed uncertainty estimation model. In this way, we can achieve sampling-free uncertainty estimation during test-time. Note that our PUA module takes the concatenation of image and BCVAE's output as the new input, to generate directly the uncertainty map.

C. Objective Function

We use the structure-aware loss function (L^{S}) [79], Kullback–Leibler divergence loss (D_{KL} in Eq. 3) and cross entropy loss (L^{CE}) for model prediction measure, latent variable distribution measure and uncertainty approximation measure, respectively (Fig. 2). L^{S} pays attention to both local and global structural similarities between ground truth (Y) and prediction. As a result, the loss function for our *PUENet* is:

$$\begin{aligned} L =& L_M^S(\mu_m^{prior}, Y) + L_A^S(\mu_a^{prior}, Y) \\ &+ L_M^S(\mu_m^{post}, Y) + L_A^S(\mu_a^{post}, Y) \\ &+ D_{KL}(z^{post}||z^{prior}) \\ &+ L^{CE}(\sigma_a^2, \sigma_e^2), \end{aligned} \tag{6}$$

where $L_M^S(\cdot)$ and $L_A^S(\cdot)$ denote the structure-aware loss for main and auxiliary mean predictions μ_m and μ_a, respectively (see Fig. 3). σ_e^2 and σ_a^2 are sampling based uncertainty and approximated uncertainty.

D. Implementation Details

We train *PUENet* using PyTorch with a maximum of 50 epochs. For fair comparison, both the training and testing images are re-scaled to 512×512 without using multi-scale or any augmentation strategy (except for SINet [3], all the competing methods in Table I are trained on multi-scale). Empirically, we set the dimension of the latent space (z) as 8. The learning rates of the BCVAE and PUA are initialized to 2.5e-5 and 1e-5 respectively. We use Adam optimizer and decrease the learning rate 10% after 40 epochs. Taking ResNet50 as an example, it took 8 hours of training with batch size 7 using a single NVIDIA GeForce RTX 2080Ti GPU. The training and testing details of our *PUENet* are presented in Algorithm 1 and 2 respectively.

IV. EXPERIMENTS

In this section, we present both qualitative and quantitative results and ablation studies of our *PUENet*.

A. Settings

Dataset: The benchmark training dataset is a combination of 3,040 images from COD10K training dataset [3] and 1,000 images from CAMO training dataset [1]. We then test our model on four benchmark testing datasets, namely CAMO

testing dataset [1] (250), COD10K testing dataset [3] (2,026), CHAMELEMON [75] (76) and NC4K (4,121) [4]. Please note that the number after each dataset indicates its size.

Evaluation Metrics: The widely used evaluation metrics include Mean Absolute Error, Mean F-measure [80], Mean E-measure [81] and S-measure [82] denoted as $\mathcal{M}$, F_β, E_ξ, S_α, respectively. Specifically, the F_β and $\mathcal{M}$ focus on the local (per-pixel) match between ground truth and prediction, while S_α pays attention to the object structure similarities. Besides, E_ξ considers both the local and global information.

$\mathcal{M}$ computes the mean absolute error between the ground truth $G \in \{0,1\}$ and a normalized prediction map $P \in [0,1]$, i.e.,

$$MAE = \frac{1}{WH}\sum_{i=1}^{W}\sum_{j=1}^{H} \mid G(i,j) - P(i,j) \mid, \tag{7}$$

where H and W denote the height and width of the given image, respectively.

F_β is defined as:

$$F_\beta = \frac{(1+\beta^2)Precision\ Recall}{\beta^2 Precision + Recall}, \tag{8}$$

where β^2 is set to 0.3, and the precision ($Precision$) and recall ($Recall$) are computed as follows:

$$Precision = \frac{|P \cap G|}{|P|}; Recall = \frac{|P \cap G|}{|G|}, \tag{9}$$

where P denotes a binary prediction, and G is the ground truth. Multiple P are computed by assigning different thresholds $\tau, \tau \in [0, 255]$ on the gray prediction map.

E_ξ is a cognitive vision-inspired metric to evaluate both the local and global similarities between two binary maps. Specifically, it is defined as:

$$E_\xi = \frac{1}{WH}\sum_{i=1}^{W}\sum_{j=1}^{H} \xi\left(G(i,j), P(i,j)\right), \tag{10}$$

where ξ represents the enhanced alignment matrix.

S_α evaluates the structural similarities between the prediction and the ground truth, which is defined as:

$$S = \alpha S_o + (1-\alpha) S_r, \tag{11}$$

where S_o and S_r denote the object and region based structure similarities, respectively. $\alpha \in [0,1]$ is set as 0.5 to assign equal weights to object-/region-based evaluation.

B. Performance Comparison

Quantitative Comparison: We compare our method with state-of-the-art (SOTA) COD models and show the quantitative performance in Table I. As most existing COD models are built upon ResNet50 backbone [72], we design ours with the same backbone, and the better performance validates our framework. Note that both LSR [4] and UJSC [58] are fully supervised multi-task learning frameworks with extra annotations. UJSC [58] uses extra saliency detection training dataset [83], and LSR [4] relies on extra camouflage localization and ranking dataset [4]. Differently, we only have access to the COD training dataset. Further, for fair comparison with SINet-V2 [14], we design a Res2Net50 [73] based framework. As a result, the better performance of ours with Res2Net50 and hybrid-ViT backbones shows the effectiveness of our *PUENet*.

Qualitative Comparison As shown in Fig. 4, when compared with the SOTA methods, our *PUENet* provides not only predictions that closest to ground truth, but also sampling-free predictive uncertainty maps which are able to approximate the sampling based uncertainties, thus inspiring future works towards explainable COD.

C. Ablation Study

We analyse learning strategy and network structure of our framework. For the former, we analyse the contribution of the MC-Dropout [47] based BNN, the CVAE [60] and Deep Ensembles [48] based alternative predictive uncertainty estimation solution. Note that all ablation studies of learning strategies are based on ResNet50 backbone. For the latter, we ablate the encoder and the proposed "SAM" module.

Uncertainty estimation strategy ablation: Following the conventional BNN for uncertainty estimation, we adopt MC-dropout [47] directly, which includes the pure BNN part of our network, and the performance is shown as "MC-Dropout" in Table II. Alternatively, we can achieve stochastic prediction with the CVAE [60] based framework, leading to "CVAE" in Table II. To achieve it, we remove the dropout module from our framework. Further, as another effective uncertainty estimation technique, we implement deep ensembles [48] by using multiple decoder heads within our framework. The performance is shown as "Deep Ensembles" in Table II. The consistent better performance of our framework compared with the alternative uncertainty estimation methods illustrates the effectiveness of our solution.

Further, as uncertainty estimation technique, [68] decomposes total uncertainty of ensemble-based framework to aleatoric uncertainty and epistemic uncertainty defining the former as mean entropy of each stochastic prediction, and the latter as mutual information of model prediction and the model parameters. Following this decomposition strategy, we compute the uncertainty maps of "MC-dropout", "CVAE" and ours (Fig. 5). It clearly shows that "MC-dropout" focuses on epistemic uncertainty (*e.g.*, challenging pixels outside the objects' boundaries and far from image center), while "CVAE" focuses on aleatoric uncertainty (*e.g.*, consistent high uncertain regions along objects' boundaries). Our method combines advantages of above two methods, leading to better uncertainty representations.

Encoder ablation: There can be flexible choices of backbone architectures for our BCVAE network, such as ResNet50 [72], Res2Net50 [73] and hybrid-ViT [74]. The results of our method based on multiple backbones are shown in Table I. Due to the effective long-range dependency modeling ability, the hybrid-ViT version of our *PUENet* achieves the best performance. Among the traditional convolutional neural network (CNN) backbones, we observe better performance of our model with Res2Net50. Besides segmentation performance, Table IV shows a detailed statistics towards the computational complexity of our BCVAE based on multiple encoding strategies.

This article has been accepted for publication in IEEE Transactions on Image Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIP.2023.3287137

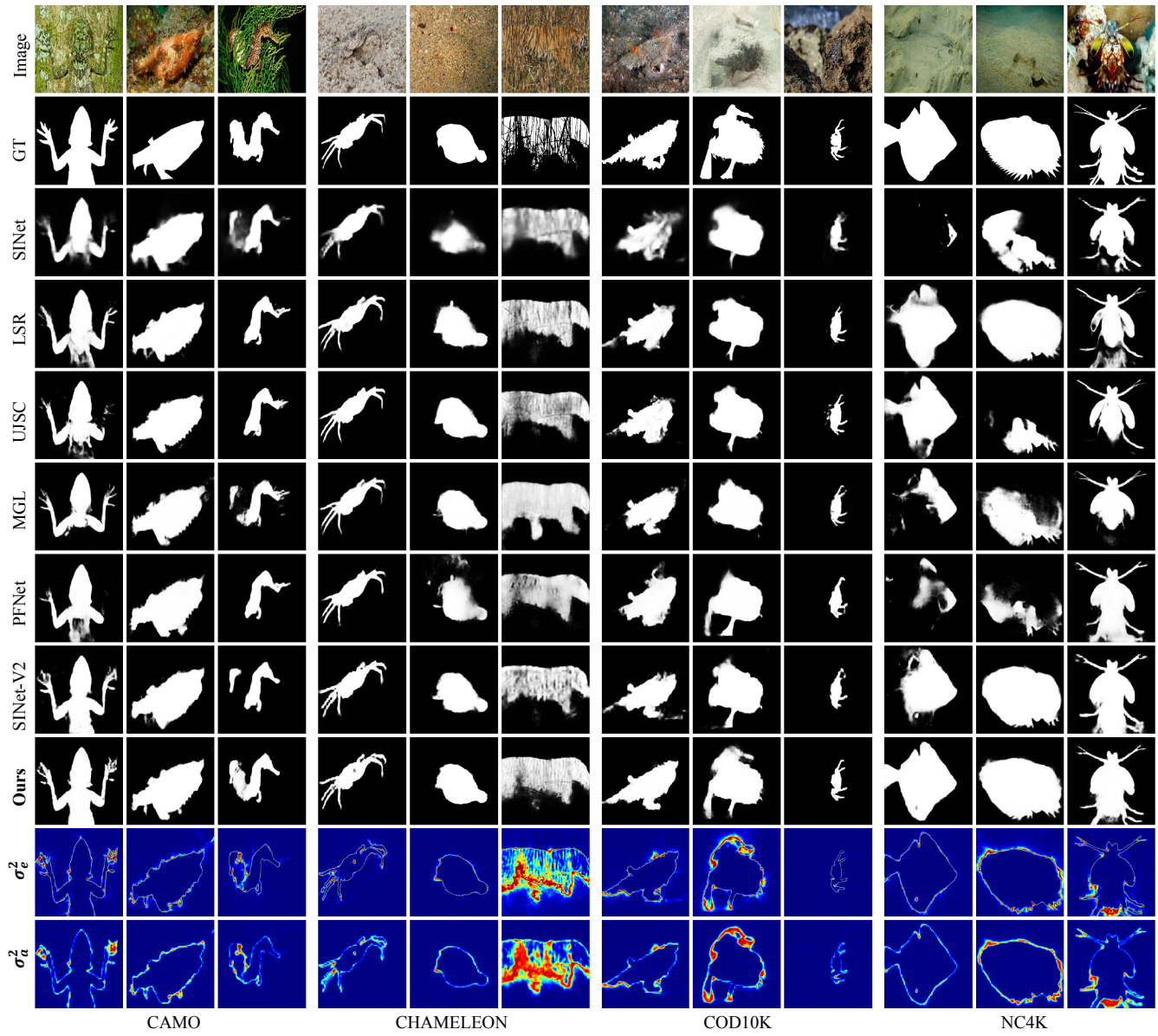

Fig. 4. Visual results of the state-of-the-art methods and our *PUENet*. "predictive uncertainty approximation" (PUA) module provides "σ_a^2", which approximates the sampling based predictive uncertainty, *i.e.*, "σ_e^2".

TABLE II
ABLATION STUDIES REGARDING DIFFERENT LEARNING STRATEGIES. ↑ INDICATES THE HIGHER THE SCORE THE BETTER, AND VICE VERSA FOR ↓.

Method	CAMO [1]				CHAMELEON [75]				COD10K [3]				NC4K [4]			
	$S_\alpha\uparrow$	$F_\beta\uparrow$	$E_\xi\uparrow$	$\mathcal{M}\downarrow$	$S_\alpha\uparrow$	$F_\beta\uparrow$	$E_\xi\uparrow$	$\mathcal{M}\downarrow$	$S_\alpha\uparrow$	$F_\beta\uparrow$	$E_\xi\uparrow$	$\mathcal{M}\downarrow$	$S_\alpha\uparrow$	$F_\beta\uparrow$	$E_\xi\uparrow$	$\mathcal{M}\downarrow$
MC-Dropout	.793	.750	.855	.082	.877	.818	.927	.035	.806	.703	.876	.039	.836	.785	.890	.051
CVAE	.785	.744	.838	.086	.881	.832	.935	.034	.807	.718	.881	.037	.833	.791	.888	.051
Deep Ensembles	.793	.743	.842	.081	.885	.832	.936	.033	.811	.717	.876	.037	**.839**	.791	.887	.051
***PUENet*(Ours)**	**.794**	**.762**	**.857**	**.080**	**.888**	**.844**	**.943**	**.030**	**.813**	**.727**	**.887**	**.035**	.836	**.798**	**.892**	**.050**

Decoder ablation: We introduce "Selective attention module" (SAM) to benefit the decoder of our BCVAE network. To test contribution of SAM, we further design an ablated version, *i.e.*, "w/o-SAM", which does not include the four hierarchical SAM modules (Fig. 3) with the backbones of ResNet50, Res2Net50 and hybrid-ViT respectively. The results (Table III) show that our SAM is able to bring consistent improvement to model performance.

V. DISCUSSION

We further analyse the COD task and its relationship to uncertainty estimation.

Task analysis: As pointed out in the Sec. I, due to the bias of preparing the COD dataset, COD has inherent data uncertainty.

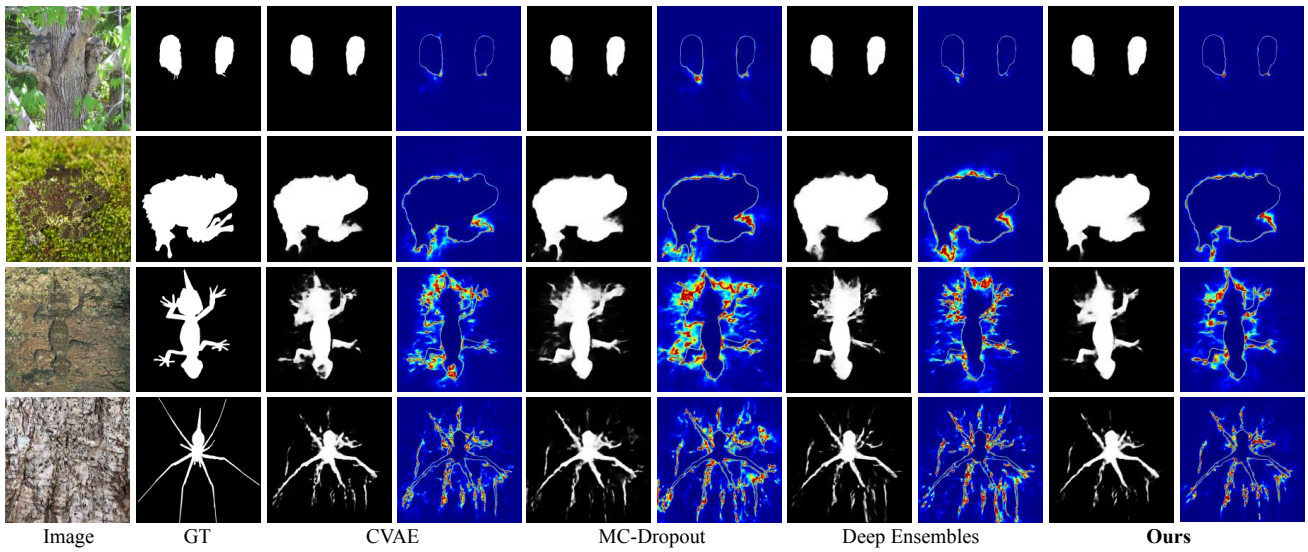

Fig. 5. Predictions and corresponding uncertainty maps of different uncertainty-aware models including "CVAE", "MC-Dropout", "Deep Ensembles" and our predictive uncertainty-based strategy ("**Ours**").

TABLE III
ABLATION STUDIES REGARDING "SELECTIVE ATTENTION MODULE" (SAM). ↑ INDICATES THE HIGHER THE SCORE THE BETTER, AND VICE VERSA FOR ↓. W-SAM, W/O-SAM DENOTE PROPOSED METHOD WITH AND WITHOUT SAM, RESPECTIVELY.

Method	Backbone	CAMO [1]				CHAMELEON [75]				COD10K [3]				NC4K [4]			
		$S_\alpha\uparrow$	$F_\beta\uparrow$	$E_\xi\uparrow$	$\mathcal{M}\downarrow$	$S_\alpha\uparrow$	$F_\beta\uparrow$	$E_\xi\uparrow$	$\mathcal{M}\downarrow$	$S_\alpha\uparrow$	$F_\beta\uparrow$	$E_\xi\uparrow$	$\mathcal{M}\downarrow$	$S_\alpha\uparrow$	$F_\beta\uparrow$	$E_\xi\uparrow$	$\mathcal{M}\downarrow$
w/o-SAM	ResNet50	.790	.747	.853	.083	.879	.820	.926	.034	.808	.707	.876	.038	.836	.787	.891	.052
w-SAM	ResNet50	.794	.762	.857	.080	.888	.844	.943	.030	.813	.727	.887	.035	.836	.798	.892	.050
w/o-SAM	Res2Net50	.829	.786	.877	.073	.887	.834	.929	.032	.836	.753	.899	.033	.857	.815	.903	.045
w-SAM	Res2Net50	.834	.806	.889	.067	.897	.858	.940	.027	.844	.774	.910	.029	.862	.830	.913	.042
w/o-SAM	Hybrid-ViT	.871	.849	.925	.046	.905	.858	.951	.024	.868	.804	.934	.023	.895	.869	.943	.030
w-SAM	Hybrid-ViT	**.877**	**.860**	**.930**	**.045**	**.910**	**.869**	**.957**	**.022**	**.873**	**.812**	**.938**	**.022**	**.898**	**.874**	**.945**	**.028**

TABLE IV
MODEL COMPLEXITY EVALUATION BASED ON DIFFERENT ENCODING STRATEGIES. THE ABBREVIATIONS IN THE TABLE ARE DETAILED AS FOLLOWS: #PARAMS = MODEL PARAMETERS. FPS = FRAME-PER-SECOND.

Model	Encoder	#Params	FPS
BCVAE	Hybrid-ViT	125.9M	19
	Res2Net50	50.9M	39
	ResNet50	48.7M	43
PUA	‡	114K	1,889

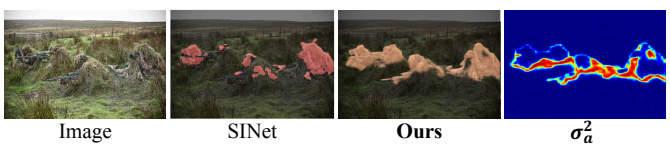

Fig. 6. Visual examples of an unseen sample from the Internet.

The bias may come from the photographers' habit of placing the targets around the center of the image, or the unavoidable stochastic noise brought by personal preference of the annotators during manual labeling process. The biased dataset based model tends to learn limited and sometimes trivial knowledge, thus generalizing poorly to samples from real-world. Another important attribute of "camouflage" is "class-agnostic", which is neglected by the existing models. Almost all the animals or humans in the wild have different levels of camouflage. However, there only exist limited categories of camouflaged instances in the current datasets. In this way, the model trained based on limited camouflage categories may learn to capture the limited category information, leading to biased model (*e.g.*, COD methods fail for detecting the snipers in Fig. 6).

Uncertainty analysis and COD: Due to the center biased dataset, the "class-agnostic" attribute and the difficulty in labeling camouflaged objects, we claim that it is necessary to perform uncertainty estimation for COD, thus avoiding over-confident predictions. Our solution of combining a BNN with a latent variable model aims to estimate both aleatoric uncertainty (indicating the labeling noise) and epistemic uncertainty (representing the biased dataset/task). As a result, our combined solution outperforms the two decomposed strategies, *i.e.*, "MC-dropout" and "CVAE" (Table II), indicating a positive effect of predictive uncertainty modeling towards improving model performance. Besides, our sampling-free uncertainty estimation technique (PUA) is effectively (σ_a^2 as shown in Fig. 4) and efficiently (1,889 FPS as presented in Table IV) used for test. In Fig. 6, we show COD model predictions with unseen image randomly collected from the Internet. Fig. 6 shows

that even when we fail to produce accurate predictions, the uncertainty-awareness (sampling-free predictive uncertainty: σ_a^2) of our solution can serve as an indicator to explain our prediction, leading to explainable COD.

Future works: [84] explains that the ImageNet pre-training may not be necessary if we train the model completely. From this view, the gap of model performance based on different backbones (*e.g.*, Table I) may not only due to the backbone structures, but also related to the convergence degree of each model. We would like to investigate it further in the future works. Besides, our method can perfectly localize the "data uncertainty" caused by labeling noise near the objects' boundaries (Fig. 4). However, "model uncertainty" that shows model ignorance of the out-of-distribution samples is still not well explored. In the future, we will explore more effective "model uncertainty" based methods, aiming for effective out-of-distribution detection.

VI. Conclusion

Considering the inherent "model bias" and "data bias" of camouflaged object detection (COD), we propose *PUENet* to achieve both accurate COD model and reliable uncertainty estimation. To reduce the sampling effort, we introduce PUA module to approximate the sampling based predictive uncertainty and achieve sampling-free uncertainty estimation during test-time. Further, we present SAM, which is exclusively designed to automatically identify and refine the challenging region predictions. Experimental results validate our solution. Importantly, the produced uncertainty map can represent our limited knowledge about this task, *i.e.*, center bias, data bias, and category bias. Although reliable uncertainty can be achieved with the proposed strategy, further investigation on uncertainty quantification and out-of-distribution sample estimation can lead to more advanced explainable COD model.

References

[1] T.-N. Le, T. V. Nguyen, Z. Nie, M.-T. Tran, and A. Sugimoto, "Anabranch network for camouflaged object segmentation," *Computer Vision and Image Understanding (CVIU)*, vol. 184, pp. 45–56, 2019. 1, 2, 6, 7, 8, 9

[2] H. Lamdouar, C. Yang, W. Xie, and A. Zisserman, "Betrayed by motion: Camouflaged object discovery via motion segmentation," in *Asian Conference on Computer Vision (ACCV)*, 2020. 1, 2

[3] D.-P. Fan, G.-P. Ji, G. Sun, M.-M. Cheng, J. Shen, and L. Shao, "Camouflaged object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2777–2787. 1, 2, 6, 7, 8, 9

[4] Y. Lv, J. Zhang, Y. Dai, A. Li, B. Liu, N. Barnes, and D.-P. Fan, "Simultaneously localize, segment and rank the camouflaged objects," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 11 591–11 601. 1, 2, 6, 7, 8, 9

[5] K. Wang, H. Bi, Y. Zhang, C. Zhang, Z. Liu, and S. Zheng, "D^{2}c-net: A dual-branch, dual-guidance and cross-refine network for camouflaged object detection," *IEEE Transactions on Industrial Electronics (TIE)*, 2021. 1, 2, 6

[6] J. Yan, T.-N. Le, K.-D. Nguyen, M.-T. Tran, T.-T. Do, and T. V. Nguyen, "Mirrornet: Bio-inspired camouflaged object segmentation," *IEEE Access*, vol. 9, pp. 43 290–43 300, 2021. 1, 2

[7] J. Zhu, X. Zhang, S. Zhang, and J. Liu, "Inferring camouflaged objects by texture-aware interactive guidance network," in *AAAI Conference on Artificial Intelligence (AAAI)*, vol. 35, 2021, pp. 3599–3607. 1, 2

[8] X. Qin, D.-P. Fan, C. Huang, C. Diagne, Z. Zhang, A. C. Sant'Anna, A. Suarez, M. Jagersand, and L. Shao, "Boundary-aware segmentation network for mobile and web applications," *arXiv preprint arXiv:2101.04704*, 2021. 1, 2

[9] J. Ren, X. Hu, L. Zhu, X. Xu, Y. Xu, W. Wang, Z. Deng, and P.-A. Heng, "Deep texture-aware features for camouflaged object detection," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 2021. 1, 2

[10] Y. Sun, G. Chen, T. Zhou, Y. Zhang, and N. Liu, "Context-aware cross-level fusion network for camouflaged object detection," *International Joint Conferences on Artificial Intelligence (IJCAI)*, 2021. 1, 2

[11] B. Dong, M. Zhuge, Y. Wang, H. Bi, and G. Chen, "Towards accurate camouflaged object detection with mixture convolution and interactive fusion," *arXiv preprint arXiv:2101.05687*, 2021. 1, 2

[12] H. Mei, G.-P. Ji, Z. Wei, X. Yang, X. Wei, and D.-P. Fan, "Camouflaged object segmentation with distraction mining," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 8772–8781. 1, 2, 3, 6

[13] Q. Zhai, X. Li, F. Yang, C. Chen, H. Cheng, and D.-P. Fan, "Mutual graph learning for camouflaged object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 12 997–13 007. 1, 2, 6

[14] D.-P. Fan, G.-P. Ji, M.-M. Cheng, and L. Shao, "Concealed object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021. 1, 2, 3, 6, 7

[15] V. Vapnik, "Principles of risk minimization for learning theory," *Advances in neural information processing systems (NeurIPS)*, vol. 4, 1991. 1

[16] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *Conference on Neural Information Processing Systems (NeurIPS)*, 2017. 1, 3, 4

[17] N. S. Detlefsen, M. Jørgensen, and S. Hauberg, "Reliable training and estimation of variance networks," in *Conference on Neural Information Processing Systems (NeurIPS)*, 2019, pp. 6326–6336. 1, 4

[18] S. Depeweg, J. M. Hernández-Lobato, F. Doshi-Velez, and S. Udluft, "Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning," in *International Conference on Machine Learning (ICML)*, 2018. 2

[19] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7132–7141. 2, 3, 5

[20] P. Bideau and E. Learned-Miller, "It's moving! a probabilistic model for causal motion segmentation in moving camera videos," in *European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 433–449. 2

[21] M. Zhang, S. Xu, Y. Piao, D. Shi, S. Lin, and H. Lu, "Preynet: Preying on camouflaged objects," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 5323–5332. 2

[22] Y. Pang, X. Zhao, T.-Z. Xiang, L. Zhang, and H. Lu, "Zoom in and out: A mixed-scale triplet network for camouflaged object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 2160–2170. 2, 6

[23] Q. Jia, S. Yao, Y. Liu, X. Fan, R. Liu, and Z. Luo, "Segment, magnify and reiterate: Detecting camouflaged objects the hard way," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 4713–4722. 2, 6

[24] X. Hu, D.-P. Fan, X. Qin, H. Dai, W. Ren, Y. Tai, C. Wang, and L. Shao, "High-resolution iterative feedback network for camouflaged object detection," *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2023. 2, 6

[25] G.-P. Ji, D.-P. Fan, Y.-C. Chou, D. Dai, A. Liniger, and L. Van Gool, "Deep gradient learning for efficient camouflaged object detection," *Machine Intelligence Research*, vol. 20, no. 1, pp. 92–108, 2023. 2, 6

[26] Z. Huang, H. Dai, S. Wang, T. Xiang, H. Chen, and J. Qin, "Feature shrinkage pyramid for camouflaged object detection with transformers," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[27] Y. Zhong, B. Li, L. Tang, S. Kuang, S. Wu, and S. Ding, "Detecting camouflaged object in frequency domain," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 4504–4513. 2, 6

[28] C. He, K. Li, Y. Zhang, L. Tang, Y. Zhang, Z. Guo, and X. Li, "Camouflaged object detection with feature decomposition and edge reconstruction," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[29] R. He, Q. Dong, J. Lin, and R. W. Lau, "Weakly-supervised camouflaged object detection with scribble annotations," *AAAI Conference on Artificial Intelligence (AAAI)*, 2023. 2

[30] T.-N. Le, Y. Cao, T.-C. Nguyen, M.-Q. Le, K.-D. Nguyen, T.-T. Do, M.-T. Tran, and T. V. Nguyen, "Camouflaged instance segmentation in-the-wild: Dataset, method, and benchmark suite," *IEEE Transactions on Image Processing (TIP)*, vol. 31, pp. 287–300, 2021. 2

[31] H. Bi, C. Zhang, K. Wang, J. Tong, and F. Zheng, "Rethinking camouflaged object detection: Models and datasets," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 2021. 2

[32] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille, "The secrets of salient object segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 280–287. 2

[33] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *IEEE International Conference on Computer Vision (ICCV)*, 2009, pp. 2106–2113. 2

[34] S. S. Kruthiventi, K. Ayush, and R. V. Babu, "Deepfix: A fully convolutional neural network for predicting human eye fixations," *IEEE Transactions on Image Processing (TIP)*, vol. 26, no. 9, pp. 4446–4456, 2017. 2

[35] S. Jetley, N. Murray, and E. Vig, "End-to-end saliency mapping via probability distribution prediction," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 5753–5761. 2

[36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *International Conference on Learning Representations (ICLR)*, 2015. 2

[37] R. Droste, J. Jiao, and J. A. Noble, "Unified image and video saliency modeling," in *European Conference on Computer Vision (ECCV)*. Springer, 2020, pp. 419–435. 2

[38] I. Jindal, M. Nokleby, and X. Chen, "Learning deep networks from noisy labels with dropout regularization," in *IEEE International Conference on Data Mining (ICDM)*. IEEE, 2016, pp. 967–972. 2

[39] K. Lee, S. Yun, K. Lee, H. Lee, B. Li, and J. Shin, "Robust inference via generative classifiers for handling noisy labels," in *International Conference on Machine Learning (ICML)*. PMLR, 2019, pp. 3763–3772. 2

[40] A. Ghosh, H. Kumar, and P. Sastry, "Robust loss functions under label noise for deep neural networks," in *AAAI Conference on Artificial Intelligence (AAAI)*, vol. 31, 2017. 3

[41] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, and L. Qu, "Making deep neural networks robust to label noise: A loss correction approach," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1944–1952. 3

[42] R. Wang, T. Liu, and D. Tao, "Multiclass learning with partially corrupted labels," *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, vol. 29, no. 6, pp. 2568–2580, 2017. 3

[43] S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich, "Training deep neural networks on noisy labels with bootstrapping," *International Conference on Learning Representations Workshop (ICLRw)*, 2015. 3

[44] Y. Li, J. Yang, Y. Song, L. Cao, J. Luo, and L.-J. Li, "Learning from noisy labels with distillation," in *IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 1910–1918. 3

[45] X. Xia, T. Liu, B. Han, C. Gong, N. Wang, Z. Ge, and Y. Chang, "Robust early-learning: Hindering the memorization of noisy labels," in *International Conference on Learning Representations (ICLR)*, 2021. 3

[46] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *International Conference on Learning Representations (ICLR)*, 2015. 3

[47] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *International Conference on Machine Learning (ICML)*. PMLR, 2016, pp. 1050–1059. 3, 4, 7

[48] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," *Conference on Neural Information Processing Systems (NeurIPS)*, 2017. 3, 7

[49] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 510–519. 3

[50] W. Qilong, W. Banggu, Z. Pengfei, L. Peihua, Z. Wangmeng, and H. Qinghua, "Eca-net: Efficient channel attention for deep convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[51] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 3–19. 3

[52] J. Park, S. Woo, J.-Y. Lee, and I. S. Kweon, "Bam: Bottleneck attention module," *British Machine Vision Conference (BMVC)*, 2018. 3

[53] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Conference on Neural Information Processing Systems (NeurIPS)*, 2017, pp. 5998–6008. 3

[54] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3146–3154. 3

[55] X. Lu, W. Wang, C. Ma, J. Shen, L. Shao, and F. Porikli, "See more, know more: Unsupervised video object segmentation with co-attention siamese networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3623–3632. 3

[56] Y. Zhang, G. Chen, Q. Chen, Y. Sun, Y. Xia, O. Deforges, W. Hamidouche, and L. Zhang, "Learning synergistic attention for light field salient object detection," *Proceedings of the British Machine Vision Conference (BMVC)*, 2021. 3

[57] F. Yang, Q. Zhai, X. Li, R. Huang, A. Luo, H. Cheng, and D.-P. Fan, "Uncertainty-guided transformer reasoning for camouflaged object detection," in *IEEE International Conference on Computer Vision (ICCV)*, 2021, pp. 4146–4155. 3, 6

[58] A. Li, J. Zhang, Y. Lv, B. Liu, T. Zhang, and Y. Dai, "Uncertainty-aware joint salient object and camouflaged object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 10 071–10 081. 3, 6, 7

[59] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020. 3

[60] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," *Conference on Neural Information Processing Systems (NeurIPS)*, vol. 28, pp. 3483–3491, 2015. 3, 4, 7

[61] J. Zhang, D.-P. Fan, Y. Dai, S. Anwar, F. S. Saleh, T. Zhang, and N. Barnes, "Uc-net: Uncertainty inspired rgb-d saliency detection via conditional variational autoencoders," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 8582–8591. 3, 5

[62] J. Zhang, D.-P. Fan, Y. Dai, S. Anwar, F. Saleh, S. Aliakbarian, and N. Barnes, "Uncertainty inspired rgb-d saliency detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021. 3

[63] P. Yan, Z. Wu, M. Liu, K. Zeng, L. Lin, and G. Li, "Unsupervised domain adaptive salient object detection through uncertainty-aware pseudo-label learning," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2022. 3

[64] Y. Wang, W. Zhang, L. Wang, T. Liu, and H. Lu, "Multi-source uncertainty mining for deep unsupervised saliency detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 11 727–11 736. 3

[65] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *IEEE International Conference on Computer Vision (ICCV)*, 2021, pp. 12 179–12 188. 5

[66] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "Denseaspp for semantic segmentation in street scenes," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 3684–3692. 5

[67] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020. 5

[68] S. Depeweg, J. M. Hernández-Lobato, F. Doshi-Velez, and S. Udluft, "Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning," in *International Conference on Machine Learning (ICML)*, 2018. 3, 7

[69] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," *Proceedings of the British Machine Vision Conference (BMVC)*, 2017. 4

[70] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *International Conference on Learning Representations (ICLR)*, 2014. 4

[71] J. Liu, J. Zhang, and N. Barnes, "Confidence-aware learning for camouflaged object detection," *arXiv preprint arXiv:2106.11641*, 2021. 5

[72] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. 5, 7

[73] S. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. H. Torr, "Res2net: A new multi-scale backbone architecture," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2019. 5, 7

[74] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly,

J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *International Conference on Learning Representations (ICLR)*, 2021. 5, 7

[75] P. Skurowski, H. Abdulameer, J. Baszczyk, T. Depta, A. Kornacki, and P. Kozie, "Animal camouflage analysis: Chameleon database." in *Unpublished Manuscript*, 2018. 6, 7, 8, 9

[76] H. Bi, C. Zhang, K. Wang, and R. Wu, "Towards accurate camouflaged object detection with in-layer information enhancement and cross-layer information aggregation," *IEEE Transactions on Cognitive and Developmental Systems*, 2022. 6

[77] Y. Sun, S. Wang, C. Chen, and T.-Z. Xiang, "Boundary-guided camouflaged object detection," *International Joint Conferences on Artificial Intelligence (IJCAI)*, 2022. 6

[78] H. Zhu, P. Li, H. Xie, X. Yan, D. Liang, D. Chen, M. Wei, and J. Qin, "I can find you! boundary-guided separated attention network for camouflaged object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2022. 6

[79] J. Wei, S. Wang, and Q. Huang, "F^3net: Fusion, feedback and focus for salient object detection," in *AAAI Conference on Artificial Intelligence (AAAI)*, vol. 34, 2020, pp. 12 321–12 328. 6

[80] R. Achanta, S. Hemami, F. Estrada, and S. Süsstrunk, "Frequency-tuned salient region detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 1597–1604. 7

[81] D.-P. Fan, G.-P. Ji, X. Qin, and M.-M. Cheng, "Cognitive vision inspired object segmentation metric and loss function," *SCIENTIA SINICA Informationis*, 2021. 7

[82] D.-P. Fan, M.-M. Cheng, Y. Liu, T. Li, and A. Borji, "Structure-measure: A new way to evaluate foreground maps," in *IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 4548–4557. 7

[83] L. Wang, H. Lu, Y. Wang, M. Feng, D. Wang, B. Yin, and X. Ruan, "Learning to detect salient objects with image-level supervision," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 136–145. 7

[84] K. He, R. Girshick, and P. Dollar, "Rethinking imagenet pre-training," in *IEEE International Conference on Computer Vision (ICCV)*, 2019. 10