Uncertainty-Guided Transformer Reasoning for Camouflaged Object Detection

Fan Yang [1,‡] Qiang Zhai[2,‡] Xin Li[1*] Rui Huang[2] Ao Luo[3] Hong Cheng[1] Deng-Ping Fan[4]
[1] AIQ [2] UESTC [3] Megvii [4] IIAI
‡ Equal contributions

Abstract

Spotting objects that are visually adapted to their surroundings is challenging for both humans and AI. Conventional generic / salient object detection techniques are suboptimal for this task because they tend to only discover easy and clear objects, while overlooking the difficult-to-detect ones with inherent uncertainties derived from indistinguishable textures. In this work, we contribute a novel approach using a probabilistic representational model in combination with transformers to explicitly reason under uncertainties, namely uncertainty-guided transformer reasoning (UGTR), for camouflaged object detection. The core idea is to first learn a conditional distribution over the backbone's output to obtain initial estimates and associated uncertainties, and then reason over these uncertain regions with attention mechanism to produce final predictions. Our approach combines the benefits of both Bayesian learning and Transformer-based reasoning, allowing the model to handle camouflaged object detection by leveraging both deterministic and probabilistic information. We empirically demonstrate that our proposed approach can achieve higher accuracy than existing state-of-the-art models on CHAMELEON, CAMO and COD10K datasets. Code is available at https://github.com/fanyang587/UGTR.

1. Introduction

Camouflaged object detection (COD), also known as decamouflaging, aims to discover the hidden targets from a given scene. It is not only an important scientific topic on understanding the relationship between visual perception and camouflage, but also can facilitate many real-life applications, such as image synthesis [4], species discovery [41] and medical image analysis [13]. However, body colours, patterns and other morphological adaptations of camouflaged object(s) would significantly decrease their probability of being detected, recognized or targeted, making decamouflaging difficult for both humans and machines.

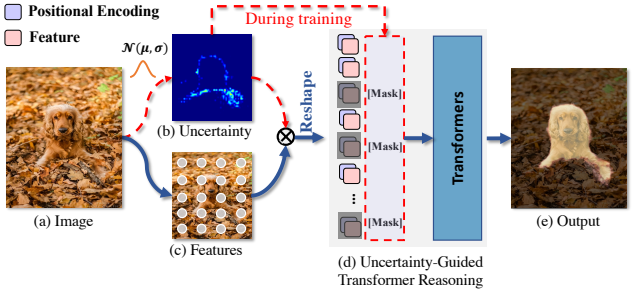

Figure 1: **Our idea** is to consider object decamouflaging as mixtures of probabilistic-deterministic procedures. The probabilistic representational model is used to capture uncertainty (dashed line), and the Transformer-based model is learned to exploit context for overcoming ambiguity guided by the mined uncertainty (solid line).

Over decades, researchers and scientists have been trying to build machine intelligence that is capable of seeing through camouflage [45, 49]. Early attempts on COD used handcrafted features in an unsupervised way, *e.g.,* colour and intensity features [21], 3D convexity [40] and motion boundary [18]. Recently, convolutional neural networks (CNNs) have been used to address the COD problem. To solve the ambiguities caused by indistinguishable textures, unlike generic / salient object detection models [9,16,31,32,35,43,59,61], current COD approaches usually distill extra features from the shared context (*e.g.,* features for identification [12], classification [29] and boundary detection [57]) and incorporate them for joint representation learning via cross-modality fusion techniques. Despite their progress, none of these approaches has explicitly taken into account the uncertainties caused by camouflage strategies, making the representation learning for COD not well-targeted and even easily misguided, not to mention the negative effects of the inherent modality difference of auxiliary- and main-task features. Empirically, we find that existing techniques cannot properly identify those true masters of camouflage which hide their outlines perfectly.

For better decamouflaging performance, we inject Bayesian learning into Transformer-based reasoning, and

*Corresponding author: Xin Li *(xinli_uestc@hotmail.com)*

propose the **U**ncertainty-**G**uided **T**ransformer **R**easoning (dubbed as **UGTR**) as a new learning paradigm. That is, we first obtain initial estimates and quantify the corresponding uncertainties with the probabilistic representational model [23, 24, 51]. Then, we reason about context to infer the *difficult-to-detect* (uncertain) regions with transformers [7, 50]. Therefore, as illustrated in Figure 1, our UGTR converts the learning of the deterministic mapping [12, 29, 55, 57] to a more complicated, uncertainty-guided context reasoning procedure. We expect that our carefully designed UGTR will be able to *reason with context information under conditions of uncertainty*, and reliably infer the concealed objects by leveraging both deterministic and probabilistic information.

More specifically, our UGTR is composed of three main components: **i)** uncertainty quantification network (UQN), **ii)** uncertainty-induced transformer (UGT) and **iii)** auxiliary prototyping transformer (PT). To obtain uncertainty maps, we draw inspiration from Bayesian probability theory [1, 23, 25, 26, 62] and design our UQN as a probabilistic representational model, which learns probability distributions rather than pixel estimates. We draw K samples from the learned distributions to produce initial estimates and measure the uncertainties. Then, our UGT comprehensively exploits richer context to infer the *difficult-to-detect* (uncertain) regions via attention mechanism. Moreover, to get the transformer focused on uncertain regions, we introduce an uncertainty-guided random masking algorithm (UGRM) that automatically assigns higher probability of *being masked out* to uncertain regions during training. Accordingly, the transformer is trained to be efficient at inferring and recovering the content in uncertain regions by exploiting context information. Last but not least, an auxiliary transformer, called Prototyping Transformer (PT), is plugged to assist UGT in mining higher-level semantics. We design multiple loss functions for our UGTR, making all components (*i.e.,* UQN, UGT and PT) being learned jointly from the raw data.

Our UGTR achieves state-of-the-art (SOTA) performance on the *CHAMELEON* [46], *CAMO* [29] and *COD10K* [12] without requiring any extra information (*e.g.,* fixation or boundary). Besides, by explicitly modeling uncertainties, our UGTR also increases the interpretability of COD models and facilitates further analysis and study of visual camouflage. Our contributions are three-fold:

- **A new learning paradigm for camouflaged object detection.** To our knowledge, this is the first attempt to introduce Bayesian learning into Transformer-based reasoning for camouflaged object detection. It converts the deterministic mapping process of conventional COD models to an uncertainty-guided context reasoning procedure. By explicitly quantifying uncertainty that carries crucial information, it enables *well-directed* context reasoning to overcome all difficulties in camouflage analysis.
- **A novel uncertainty-guided transformer reasoning model for camouflaged object detection.** We present the uncertainty-guided transformer reasoning model (UGTR) that integrates all novel components, such as uncertainty quantification network (UQN), prototyping transformer (PT) and uncertainty-guided transformer (UGT), within a unified, end-to-end framework for camouflaged object detection. It should be noted that our proposed UGRM algorithm serves as a *hard-example-mining* module, which uses uncertainty guidance to enhance UGT's context reasoning capability during training.
- **State-of-the-art results on widely-used benchmarks.** Our fully-equipped UGTR achieves SOTA performance on a variety of benchmarks, including *CHAMELEON* [46], *CAMO* [29] and *COD10K* [12], and outperforms existing COD models by a large margin.

2. Related Work

Camouflaged Object Detection. Camouflaged object detection (COD), as a task of distinguishing the camouflaged target from its background, has been put into wide application. Pioneer works use handcrafted features to discriminate objects from the background in an unsupervised manner, such as colour and intensity features [21], 3D convexity [40], and motion boundary [18]. Recently, deep learning based approaches try to address the COD problem in a data-driven manner and have achieved impressive results in identifying / detecting camouflaged objects. To better handle indistinguishable textures (or boundaries), the existing approaches focus on exploring auxiliary information, *e.g.,* fixations [36], boundaries [57], image-level labels [12, 29], for joint representation learning. Unlike these methods, we combine the benefits of Bayesian learning and Transformer-based reasoning as a new learning paradigm. Our approach reformulates the mainstream deterministic mapping process into a more reliable, uncertainty-guided context reasoning procedure, which sets new records on all benchmarks.

Bayesian Deep Learning. Bayesian neural networks (BNN) [1, 15, 24, 37, 44] are well-known for modeling uncertainties in neural networks. The key idea of BNN is to learn the distribution over network weights [1] or features [56] instead of outputing a single fixed value. Notable works on Bayesian approaches for modern deep learning include [1, 15, 24, 37, 44]. Recently, Gal *et.al.* [14, 15] have cast *dropout* as approximate Bayesian inference over network weights. Kendall *et.al.* [23] show that a posterior distribution of pixel-class labels can be generated by Monte Carlo sampling with dropout at test time. These techniques have been successfully applied to modeling uncertainty for semantic segmentation / scene parsing [19, 23], person re-identification [56] and medical image analysis [28]. In-

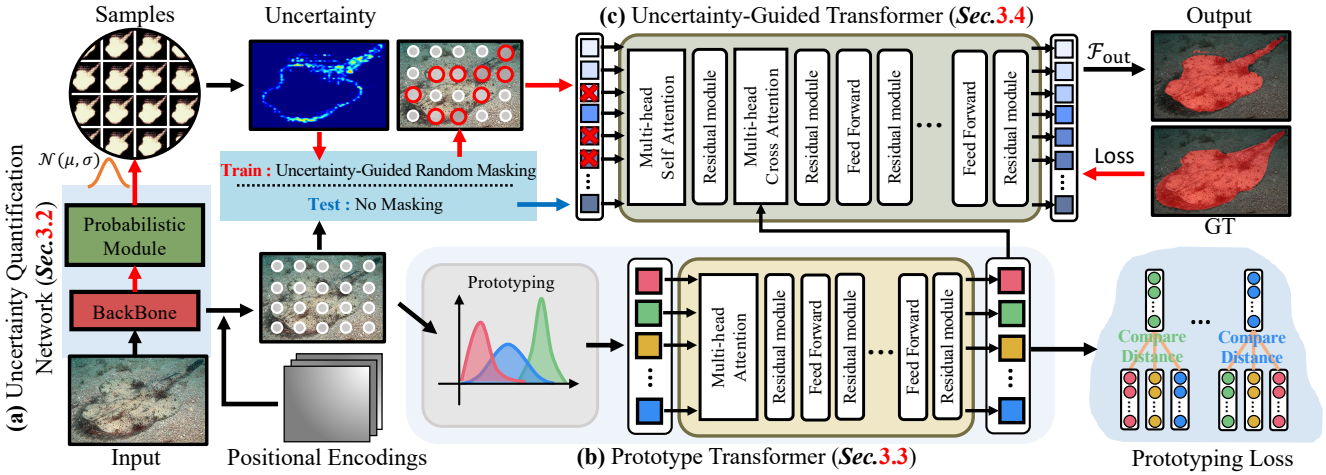

Figure 2: **An overview of our proposed uncertainty-guided transformer reasoning model (UGTR)**. Our UGTR includes three main components, *i.e.,* UQN, PT and UGT, marked by (a)-(c). Please refer to § 3 for more details. Best viewed in color.

spired by these works, we build our uncertainty quantification network (UQN) as a probabilistic representational model to capture the uncertainty for camouflaged object detection. To the best of our knowledge, this is the first attempt to explicitly model uncertainty and fully leverage it to improve the reliability of camouflaged object detection.

Visual Transformers. Transformer was first introduced by [50] to handle sequential data in the field of machine translation. Recent works have tried to apply transformers to various vision tasks, such as object detection [2, 5, 66], image recognition [8, 47], multi-object tracking [39, 48], semantic segmentation [64], and human pose and mesh reconstruction [30]. These works have proven that images can be learned in a sequence-to-sequence manner. Importantly, the random masking technique enforces the transformer to reason over context for inferring the masked contents with attention mechanism (*i.e.,* multi-head attention), largely improving the model's reasoning power. In this paper, we propose two novel transformers — uncertainty-guided transformer (UGT) and prototyping transformer (PT). UGT reasons over context to achieve pixel-wise predictions while PT works as an auxiliary transformer to mine high-level semantics. Importantly, a novel uncertainty-guided random masking (UGRM) algorithm is introduced which works as a bridge connecting our uncertainty quantification and transformer reasoning so that UGT is learned to focus on difficult (uncertain) regions during training.

3. Our Approach

3.1. Preliminaries

Task Setup and Notations. Following [12, 29, 36, 57], we treat COD as a class-independent, pixel-wise segmentation task. Formally, let $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$ and $\mathbf{C} \in \mathbb{R}^{H\times W\times 1}$ denote the input image and output camouflage map, respectively. Given a large collection of such pairs $\{\mathbf{I}_i, \mathbf{C}_i\}_{\mathbf{i}=1}^{N}$, our task is to learn a mapping function $\mathcal{F}_\Theta$ parameterized by weights Θ that can correctly transfer the novel input to its corresponding camouflage map. For each pixel (position) $p_o \in [1, H \times W]$, the estimated score $c_{p_o} \in [0, 1]$ reflects the COD model's prediction, where a score of '1' indicates that it belongs to the camouflaged objects and vice versa.

Our Idea. Unlike prior arts [12, 29, 36, 57] that simply consider the mapping $\mathcal{F}_\Theta$ as a deterministic process, we formulate it as the mixtures of probabilistic-deterministic procedures. We argue that the decamouflaging process, even for human perception, is usually full of uncertainties, and thus modeling the COD problem requires the use of both probabilistic and deterministic information for reasoning.

Approach Overview. To verify our idea, we present a novel **U**ncertainty-**G**uided **T**ransformer **R**easoning model (**UGTR**). As shown in Figure 2, UGTR includes three major components:

- **Uncertainty Quantification Network (§ 3.2).** Our uncertainty quantification network (UQN), denoted as $\mathcal{F}_\theta$, contains two parts: feature extractor (backbone) $\mathcal{F}_{\theta 1}$ and probabilistic module $\mathcal{F}_{\theta 2}$. For $\mathcal{F}_{\theta 1}$, it takes the RGB image $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$ as input, and produces feature embeddings $\mathbf{F} \in \mathbb{R}^{h\times w\times c}$: $\mathbf{I} \overset{\mathcal{F}_{\theta 1}}{\rightarrow} \mathbf{F}$. Then, $\mathcal{F}_{\theta 2}$ models the variance of backbone's output as a measure of uncertainty. Following [24, 56], we model the *pixel-wise* distribution as Gaussian parameterized by mean map $\boldsymbol{\mu} \in \mathbb{R}^{h\times w\times 1}$ and variance map $\boldsymbol{\sigma} \in \mathbb{R}^{h\times w\times 1}$: $\mathbf{F} \overset{\mathcal{F}_{\theta 2}}{\rightarrow} (\boldsymbol{\mu}, \boldsymbol{\sigma})$, and draw K samples (camouflage maps) from the distributions to produce the uncertainty map.
- **Prototyping Transformer (§ 3.3).** Our prototyping transformer (PT) $\mathcal{F}_\delta$ works as an auxiliary module to learn and reason over higher-level semantics. It transforms $\mathbf{F}$ to t semantic prototypes: $\mathbf{F} \overset{\mathcal{F}_\delta}{\rightarrow} \mathbf{X}$, where $\mathbf{X} = \{x_1, \ldots, x_t\} \in \mathbb{R}^{t\times c}$ denotes the set of the learned prototypes. We expect that our comprehensively mined

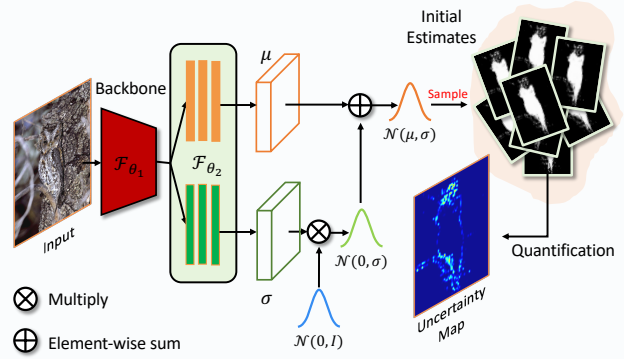

Figure 3: **Illustration of UQN**. UQN works as a probabilistic model for uncertainty quantification, which is composed of a backbone network and a probabilistic module. Please refer to § 3.2 for details.

$\mathbf{X}$ can be used to assist final reasoning.

- **Uncertainty-Guided Transformer (§ 3.4).** Finally, our uncertainty guided transformer (UGT) $\mathcal{F}_\phi$ takes $\mathbf{F}$, $\mathbf{X}$ and $\mathbf{U}$ as inputs, and produces the refined features $\breve{\mathbf{F}} \in \mathbb{R}^{(h\times w)\times c}$ for COD in a sequence-to-sequence manner: $(\mathbf{F}, \mathbf{X}, \mathbf{U}) \overset{\mathcal{F}_\phi}{\rightarrow} \breve{\mathbf{F}}$. To enhance the context reasoning capability, especially for difficult (uncertain) regions, we introduce an uncertainty-guided random masking algorithm (UGRM). It is embedded in UGT and guided by $\mathbf{U}$ to ensure the *difficult-to-detect* regions are easier to be masked out during training. Therefore, UGT is trained to enhance its capability of inferring the uncertain information (by fully exploring contextual features).

All components above work together to fully reason over the context and deliver the final representation $\breve{\mathbf{F}}$ so that the entire decamouflaging process involves both probabilistic and deterministic procedures. To achieve the final predictions, $\breve{\mathbf{F}}$ can be easily mapped to $\mathbf{C}$ with a readout function $\mathcal{F}_{out}$: $\breve{\mathbf{F}} \overset{\mathcal{F}_{out}}{\rightarrow} \mathbf{C}$, which is composed of a reshape layer, a 1×1 convolutional layer and upsampling layer. In the following sections, we will detail each major component.

3.2. Uncertainty Quantification Network

Distribution Modeling. We build our uncertainty quantification network (UQN) $\mathcal{F}_\theta$ as a probabilistic representational model to measure uncertainty. Therefore, in this stage, what $\mathcal{F}_\theta$ delivers for each pixel (*e.g.* the pixel p) is a distribution parameterized by mean $\boldsymbol{\mu}_p$ and variance $\boldsymbol{\sigma}_p$ instead of a scalar (*e.g.,* a score). Following [23], we model the distribution of outputs at each pixel as Gaussian. That is to say we think of our UQN's prediction as a random variable. We expect that the camouflage score at the position p can be drawn from the learned distribution: $c_p \sim \mathcal{N}(\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p)$, where $\boldsymbol{\mu}_p$ and $\boldsymbol{\sigma}_p$ are produced by UQN. As already observed by existing works [15, 23, 24, 56], when we use random sample to train $\mathcal{F}_\theta$, the error will not be propagated back from the output. To solve this problem, inspired by [26], we decompose the direct sampling operation into a trainable part and a random part. Specifically, we first draw a sample ϵ_p from the standard Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$ randomly, *i.e.,* $\epsilon_p \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and then obtain the sample by computing $\boldsymbol{\mu}_p + \epsilon_p \boldsymbol{\sigma}_p$. By doing so, the gradients can be propagated backward to optimize the parameters θ.

Network Architecture. To instantiate the formulation above by neural network, we design our UQN by building two separate branches upon the underlying feature extractor (backbone) for $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$, respectively. As shown in Figure 3, our UQN includes two major parts: backbone $\mathcal{F}_{\theta 1}$ and probabilistic module $\mathcal{F}_{\theta 2}$. Concretely, given the input image $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$, the backbone $\mathcal{F}_{\theta 1}$ is employed to obtain c-dimensional feature embeddings: $\mathbf{F} = \mathcal{F}_{\theta 1}(\mathbf{I}) \in \mathbb{R}^{h\times w\times c}$. Then, a two-branch probabilistic module $\mathcal{F}_{\theta 2}$ further transfers $\mathbf{F}$ to $\boldsymbol{\mu}_p$ and $\boldsymbol{\sigma}_p$:

$$\boldsymbol{\mu} = \mathcal{F}_{\theta 2_\mu}(\mathbf{F}), \quad \boldsymbol{\sigma} = \mathcal{F}_{\theta 2_\sigma}(\mathbf{F}), \tag{1}$$

where $\boldsymbol{\mu} \in \mathbb{R}^{h\times w\times 1}$ and $\boldsymbol{\sigma} \in \mathbb{R}^{h\times w\times 1}$ denote the mean map and variance map, respectively. Moreover, a standard Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$ is attached to make the entire process end-to-end differentiable.

Distribution Learning Loss $\mathcal{L}_{\mathbf{DLL}}$. To train our UQN, we design the following loss function $\mathcal{L}_{\mathbf{DLL}}$, which is a weighted combination of a standard BCE loss and a Kullback-Leibler (KL) divergence [26]:

$$\mathcal{L}_{\mathbf{DLL}} = \mathcal{L}_{\mathbf{BCE}}(c^{(k)}, \mathbf{C}_{gt}) + \eta \cdot \mathcal{D}(\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}) \parallel \mathcal{N}(\mathbf{0}, \mathbf{I})) \tag{2}$$

where η means the combination weight which is set to be 0.1 to emphasize model's prediction, $c^{(k)}$ means a sample randomly drawn from the learned distribution and $\mathbf{C}_{gt}$ denotes the ground truth [1].

Uncertainty Quantification. To measure pixel-wise uncertainty, we can sample K initial camouflage maps from the learned distribution, denoted as $C_{init} = \{c^{(1)}, \ldots, c^{(K)}\}$. According to Bayesian probability theory [1, 23, 25, 26], we can simply treat C_{init} as empirical samples from an approximate predictive distribution and measure how confident the model is in its prediction by computing the variance:

$$\mathbf{U} = \mathrm{Norm}(\mathrm{Var}(\mathrm{C_{init}})), \tag{3}$$

where $\mathbf{U} \in \mathbb{R}^{h\times w\times 1}$ means the uncertainty map, $\mathrm{Norm}(\cdot)$ is the mean-max normalization operation and $\mathrm{Var}(\cdot)$ denotes the operation of computing variance.

3.3. Prototyping Transformer

In addition to the uncertainty, the high-level, global context information also plays a critical role in decamouflaging. However, for the task of camouflaged object detection,

[1] To promote diversity, we only draw one example to compute the loss.

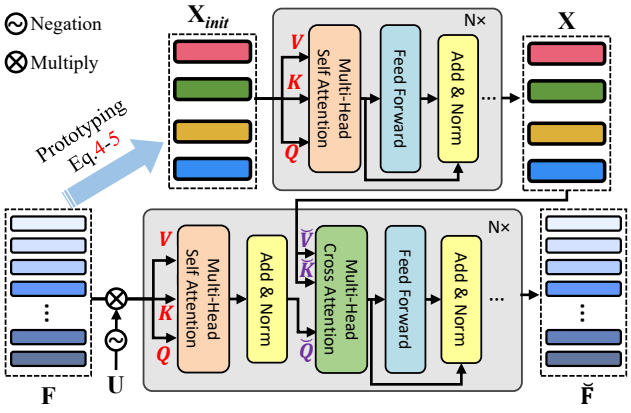

Figure 4: **Illustration of PT and UGT**. PT captures high-level semantics while UGT reasons over underlying features to learn the final representations for COD. For brevity, positional encodings are not included. Please refer to § 3.3 and § 3.4 for details.

the conventional techniques, such as global average pooling, are not reliable due to texture similarities between camouflaged objects and the background. Thus, we tackle this problem using metric learning, and design a prototyping transformer (PT), denoted as $\mathcal{F}_\delta$, to obtain representative and discriminative prototypes $\mathbf{X} = \{x_1, \ldots, x_t\} \in \mathbb{R}^{t\times c}$.

Generally, our PT is implemented in a visual transformer framework [8, 47], as shown in Figure 4 (Top). The novelty of PT lies in its prototyping procedure under the supervision of a prototyping loss $\mathcal{L}_{\mathbf{PL}}$. Importantly, unlike ACT [63] that groups the features using Locality Sensitive Hashing (LSH) to reduce the computation cost, our PT clusters all features in an iterative manner to overcome ambiguity.

Prototyping. Since our PT is based on transformer architecture, given feature embeddings $\mathbf{F} \in \mathbb{R}^{h\times w\times c}$, we first collapse the spatial dimensions of $\mathbf{F}$ into one dimension, *i.e.,* a $c \times hw$ feature map, and encode the positional information with fixed positional encodings [2] (denoted as $\mathbf{P}_F = \{p_{f_1}, \ldots, p_{f_{hw}}\}$). Then, let us denote $\mathbf{V} = \{v_1, \ldots, v_t\}$ as a set of t learnable visual atoms stored in an external memory. Inspired by the expectation–maximization (**EM**) algorithm [6], we employ an iterative strategy to get the initial prototypes $\mathbf{X}_{init}$. That is, we first fix $\mathbf{X}_{init}$ (initially, we set $\mathbf{X}^{(0)}_{init} = \mathbf{V}$), and compute the affinity map $\mathbf{A}$ by:

$$\mathbf{A}_{p,t} = \frac{e^{\kappa x_t^{\mathrm{T}} f_p}}{\sum_{t=1}^{T} e^{\kappa x_t^{\mathrm{T}} f_p}}, \tag{4}$$

where $f_p \in \mathbf{F}$ denotes the p-th feature sample, $x_t \in \mathbf{X}$ means the t-th prototype, and κ denotes the concentration parameter. Then, we fix $\mathbf{F}$ and update the prototypes by:

$$x_t = \frac{\sum_{p=1}^{hw} \mathbf{A}_{p,t} f_p}{\sum_{p=1}^{hw} \mathbf{A}_{p,t}}. \tag{5}$$

The two operations above (Eq. 4 and Eq. 5) will be repeated for multiple times to find the cluster centroids as $\mathbf{X}_{init}$. It should be noted that the visual atoms $\mathbf{V}$ are also optimized through back-propagation training.

Multi-Head Self-Attention. After the previous step, we can get t initial prototypes $\mathbf{X}_{init}$ which represent the global context of the given scene. To make these prototypes more discriminative, we follow [66] to use multiple transformer blocks [2] for representation enhancement based on multi-head (self) attention:

$$\begin{aligned} &Q_i = \mathbf{X}_{init}\mathcal{W}_i^Q, K_i = \mathbf{X}_{init}\mathcal{W}_i^K, V_i = \mathbf{X}_{init}\mathcal{W}_i^V, \\ &Head_i = Attention(Q_i, K_i, V_i), \\ &MH(Q, K, V) = Concact(Head_1, \ldots, Head_M), \end{aligned} \tag{6}$$

where $\mathcal{W}_i^Q$, $\mathcal{W}_i^K$ and $\mathcal{W}_i^V$ are learnable parameters for the i-th head, and Q_i, K_i and V_i denote the *query*, *key*, and *value* features, respectively. $M = 8$ heads are used in our implementation. Note that we take into account the positional information of each prototype by re-using the affinity map $\mathbf{A}$ to compute prototype-level postion features $\mathbf{P}_X$ by:

$$p_{x_t} = \frac{\sum_{p=1}^{hw} \mathbf{A}_{p,t} p_{f_p}}{\sum_{p=1}^{hw} \mathbf{A}_{p,t}}. \tag{7}$$

The encoded prototype positions are added to the input of each multi-head self attention layer to learn the final prototype features $\mathbf{X} \in \mathbb{R}^{c\times T}$ which carry important global semantic information.

Prototyping Loss $\mathcal{L}_{\mathbf{PL}}$. We find it helpful to use an auxiliary loss function in PT during training to push prototypes away from each other. Therefore, we define the following prototyping loss $\mathcal{L}_{\mathbf{PL}}$:

$$\mathcal{L}_{\mathbf{PL}} = \sum_{x_i, x_j \in \mathbf{X}} \max((m - dist(x_i, x_j)), 0), \tag{8}$$

where m is a pre-set threshold. $\mathcal{L}_{\mathbf{PL}}$ works as an unsupervised objective function to train our PT.

3.4. Uncertainty-Guided Transformer

Up to this point, we have obtained the initial features $\mathbf{F}$ (§ 3.2), uncertainty (difficulty) map $\mathbf{U}$ (§ 3.2) and the discriminative prototypes $\mathbf{X}$ (§ 3.3). Now, we want to use all these information to learn final representations for COD. To achieve this goal, we design a novel module, *i.e.,* the Uncertainty-Guided Transformer (UGT), to make full use of all information and deliver the final representations. Generally, as shown in Figure 4 (Bottom), our UGT is in compliance with the standard architecture of the transformer, which transforms initial $\mathbf{F}$ using multi-head self- and cross-attention mechanisms to process features $\breve{\mathbf{F}}$. Formally, our

[2] Each transformer block consists of a multi-head self-attention module and a feed forward network (FFN).

Methods	CHAMELEON [46]				CAMO-Test [29]				COD10K-Test [12]			
	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$
2017 FPN † [31]	0.794	0.783	0.590	0.075	0.684	0.677	0.483	0.131	0.697	0.691	0.411	0.075
2017 MaskRCNN † [16]	0.643	0.778	0.518	0.099	0.574	0.715	0.430	0.151	0.613	0.748	0.402	0.080
2017 PSPNet † [58]	0.773	0.758	0.555	0.085	0.663	0.659	0.455	0.139	0.678	0.680	0.377	0.080
2018 UNet++ † [65]	0.695	0.762	0.501	0.094	0.599	0.653	0.392	0.149	0.623	0.672	0.350	0.086
2018 PiCANet † [34]	0.769	0.749	0.536	0.085	0.609	0.584	0.356	0.156	0.649	0.643	0.322	0.090
2019 MSRCNN † [20]	0.637	0.686	0.443	0.091	0.617	0.669	0.454	0.133	0.641	0.706	0.419	0.073
2019 PoolNet † [33]	0.776	0.779	0.555	0.081	0.702	0.698	0.494	0.129	0.705	0.713	0.416	0.074
2019 BASNet † [42]	0.687	0.721	0.474	0.118	0.618	0.661	0.413	0.159	0.634	0.678	0.365	0.105
2019 PFANet † [60]	0.679	0.648	0.378	0.144	0.659	0.622	0.391	0.172	0.636	0.618	0.286	0.128
2019 CPD † [54]	0.853	0.866	0.706	0.052	0.726	0.729	0.550	0.115	0.747	0.770	0.508	0.059
2019 HTC † [3]	0.517	0.489	0.204	0.129	0.476	0.442	0.174	0.172	0.548	0.520	0.221	0.088
2019 EGNet † [59]	0.848	0.870	0.702	0.050	0.732	0.768	0.583	0.104	0.737	0.779	0.509	0.056
2019 ANet-SRM [29]	‡	‡	‡	‡	0.682	0.685	0.484	0.126	‡	‡	‡	‡
2020 MirrorNet [55]	‡	‡	‡	‡	0.741	0.804	0.652	0.100	‡	‡	‡	‡
2020 PraNet [13]	0.860	0.898	0.763	0.044	0.769	0.833	0.663	0.094	0.789	0.839	0.629	0.045
2020 SINet [12]	0.869	0.891	0.740	0.044	0.751	0.771	0.606	0.100	0.771	0.806	0.551	0.051
UGTR (ours)	**0.888**	**0.918**	**0.796**	**0.031**	**0.785**	**0.859**	**0.686**	**0.086**	**0.818**	**0.850**	**0.667**	**0.035**

Table 1: **Quantitative results on different benchmark datasets**. '†' means SOTA methods for GOD and SOD. ↑ (or ↓) indicates that the higher (or the lower) the better. The best scores are highlighted in **bold**.

UGT $\mathcal{F}_\phi$ takes $\mathbf{F} \in \mathbb{R}^{c\times hw}$, $\mathbf{U} \in \mathbb{R}^{1\times hw}$ and $\mathbf{X} \in \mathbb{R}^{c\times T}$ as inputs and produces $\breve{\mathbf{F}} \in \mathbb{R}^{c\times hw}$ as output. The main difference from PT is that we feed the uncertainty-aware features $\mathbf{F}_U = \mathbf{F} \otimes (1-\mathbf{U})$ to transformers ($\otimes$ element-wise multiply) and add a cross-attention layer in each transformer block so that $\mathbf{X}$ can be incorporated for representation learning:

$$\begin{aligned} &\breve{Q}_i = \mathbf{F}_U \mathcal{W}^{\breve{Q}_i}, \breve{K}_i = \mathbf{X}\mathcal{W}^{\breve{K}_i}, \breve{V}_i = \mathbf{X}\mathcal{W}^{\breve{V}_i}, \\ &\breve{Head}_i = Attention(\breve{Q}_i, \breve{K}_i, \breve{V}_i), \\ &\breve{MH}(Q,K,V) = Concact(\breve{Head}_1, \ldots, \breve{Head}_M), \end{aligned} \tag{9}$$

where $\breve{Q}_i$, $\breve{K}_i$ and $\breve{V}_i$ are learnable parameters. The high-level semantics are encoded by the cross-attention layer to assist in *pixel-wise* representation learning.

Uncertainty-Guided Random Masking. We introduce an uncertainty-guided random masking (UGRM) algorithm to induce our UGT to focus on uncertain (difficult) regions during training. Instead of adopting the widely-used random masking technique, we assign higher probabilities to those *difficult-to-detect* regions to be masked out during training guided by $\mathbf{U}$. Formally, let us denote $\mathbf{R} \in \mathbb{R}^{1\times hw}$ as a random probability map, where $r_p \in [0,1]$. We mask out those features $f_p \in \mathbf{F}_U$ whose associated uncertainty score u_p is higher than r_p: $u_p > r_p$. UGRM is more reliable than the widely-used random masking strategy [50], because it increases the difficulty and diversity of training samples. With this means, the uncertainty information has been also incorporated into the learning procedure, enabling our UGT to better deal with difficult (uncertain) regions.

Finally, the output of UGT ($\breve{\mathbf{F}}$) is mapped to final predictions by using a simple readout function $\mathcal{F}_{out}$, which is composed of a reshape layer, a 1×1 convolutional layer and upsampling layer. The loss function for our fully-equipped UGTR is a combination of multiple loss functions:

$$\mathcal{L}_{\mathbf{UGTR}} = \mathcal{L}_{\mathbf{DLL}} + \eta\mathcal{L}_{\mathbf{PT}} + \omega\mathcal{L}_{\mathbf{BCE}} \tag{10}$$

where η and ω mean the combination weights, and $\mathcal{L}_{\mathbf{BCE}}$ is the standard BCE loss for UGT.

4. Experiments

4.1. Experimental Settings

Dataset. Three public benchmark datasets are used for performance evaluation. CHAMELEON [46] is a small dataset, which is a collection of 76 high-resolution images with pixel-level labels. CAMO [29] includes a total of $2,500$ images of both naturally and artificially camouflaged objects under 8 categories. COD10K [12] is the most challenging COD dataset as it includes $10,000$ images with 10 super-classes and 78 sub-classes, where both image-wise level and pixel-wise level annotations are provided. In our approach, only the pixel-wise labels are used for model training. Following [12], we build our `train set` as a combination of the train sets from CAMO and COD10K.

Evaluation Metric. Following [12,29,36], four commonly-agreed evaluation metrics are used in our experiments: mean absolute error (MAE), mean E-measure (E_ϕ) [11], mean S-measure (S_α) [10] and F-measure (F_β^w) [38].

Training Settings. During training, the underlying backbone $\mathcal{F}_{\theta1}$ is initialized by ResNet-50 [17] pre-trained on ImageNet [27], and the remaining modules (*i.e.,* probabilistic module $\mathcal{F}_{\theta2}$, prototyping transformer $\mathcal{F}_\delta$ and uncertainty-guided transformer $\mathcal{F}_\phi$) are randomly initialized. Following regular practice [12,29], we augment each training sample with random cropping, left-right flipping and scaling in the range of $[0.75, 1.25]$. We train our UGTR model using the Stochastic Gradient Descent (SGD) with 'poly' learning rate policy: $lr = base\_lr \times (1 - \frac{iter}{max_{iter}})^{power}$. We empirically set the base learning rate $base\_lr$ to 10^{-7} and $power$ to 0.9. Note that, our UGRM is only used in the training phase, which enhances UGT's capability of reasoning over context to better handle the ambiguity in uncertain regions.

Testing Settings. Once trained, our UGTR can be applied

Candidate				CHAMELEON [46]				CAMO-Test [29]				COD10K-Test [12]			
ResNet50	UQN	UGT	PT	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$
✔				0.767	0.799	0.535	0.094	0.742	0.786	0.538	0.130	0.729	0.692	0.436	0.079
	✔			0.775	0.837	0.543	0.085	0.728	0.806	0.507	0.126	0.730	0.738	0.417	0.072
	✔	✔		0.833	0.891	0.762	0.038	0.747	0.836	0.643	0.098	0.789	0.826	0.620	0.042
	✔	✔	✔	**0.888**	**0.918**	**0.796**	**0.031**	**0.785**	**0.859**	**0.686**	**0.086**	**0.818**	**0.850**	**0.667**	**0.035**

Table 2: **The ablation results** of our proposed approach on CHAMELEON, CAMO `test` and COD10K `test`.

Method	CAMO-Test [29]				COD10K-Test [12]			
	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$
UGT + GAP	0.735	0.823	0.633	0.101	0.776	0.815	0.613	0.050
UGT + MSP [58]	0.758	0.841	0.642	0.098	0.792	0.832	0.631	0.041
UGTR (UGT + PT)	0.771	0.840	0.669	0.088	0.811	0.842	0.651	0.037
UGTR (UGT + PT + $\mathcal{L}_{\mathtt{PL}}$)	**0.785**	**0.859**	**0.686**	**0.086**	**0.818**	**0.850**	**0.667**	**0.035**

Table 3: **The comparison** using different methods for capturing high-level semantics. 'GAP' means the global average pooling method, 'MSP' means the multi-scale pooling strategy, 'PT' denotes our prototyping transformer model.

Method	CAMO-Test [29]				COD10K-Test [12]			
	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$
UGTR w/o Mask	0.761	0.842	0.646	0.095	0.797	0.836	0.640	0.040
UGTR w/ RM [50]	0.775	0.849	0.655	0.092	0.805	0.840	0.651	0.039
UGTR w/ UGRM	**0.785**	**0.859**	**0.686**	**0.086**	**0.818**	**0.850**	**0.667**	**0.035**

Table 4: **The ablation results** of different masking algorithms on CAMO `test` and COD10K `test`.

to unseen images. It should be noted that the embedded UGRM is removed in this stage. Both uncertainty map and camouflage map can be generated by our model.

4.2. Main Results

We compare UGTR with previous SOTA methods including top-ranked generic object detection (GOD) and salient object detection (SOD) approaches and recently published methods for COD.

CHAMELEON [46]. In Table 1 (left), we compare our UGTR against 14 SOTA approaches on four commonly-used metrics. The results show that GOD and SOD models tend to perform worse than customized COD models. It is because conventional generic / salient object detection techniques are suboptimal to discover *difficult-to-detect* objects. Moreover, although the existing COD approaches achieve promising results, our UGTR surpasses all the competitors by a large margin. For instance, UGTR outperforms SINet [12] in terms of MAE, mean S-measure [10], mean E-measure [11], and weighted F-measure [38] by 29.5%, 2.2%, 3.0% and 7.6%, respectively. Such a performance gain proves the importance of reasoning over the uncertain regions. Even without using any extra information, it can dramatically boost reliability and overall performance.

CAMO [29]. The quantitative comparison with 16 SOTA approaches on CAMO `test` is summarized in Table 1 (middle). We find that our UGTR again achieves the best performance across all metrics. The high accuracy should be attributed to the transformer-aided explicit reasoning, because it fully leverages the context information to learn representations and infer hidden objects. We would also like to mention that our UGTR does not require extra information, such as image-level classification information [29].

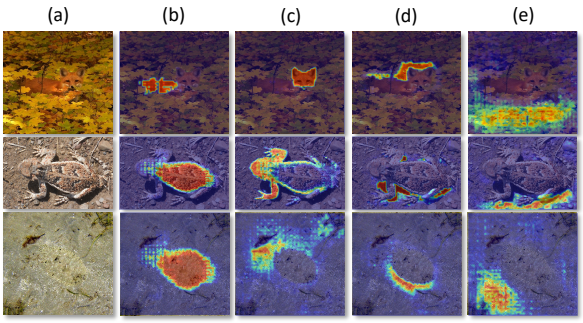

Figure 5: **Prototype visualization**. (a) Images; (b)-(e) Visualizations of different prototypes. These learned prototypes capture the high-level semantics, *e.g.,* object parts (b)-(c), boundaries (d) and background regions (e).

COD10K [12]. Table 1 (right) compares our UGTR with 14 SOTA models on the most challenging COD10K `test`. The comparison demonstrates that our UGTR can deliver reliable results. Although it is difficult to make improvement on this dataset, our novel learning paradigm still can help to overcome ambiguity and identity the camouflaged object more easily. Visual comparison results on COD10K `test` are provided in Figure 6.

4.3. Ablation Study

Effectiveness of Each Module. To verify the effectiveness of each novel module, we provide a baseline model for comparison, *i.e.,* the ResNet50-FCN. First, as shown in Table 2, our UQN achieves similar performance as ResNet50-FCN. Note that because our UQN is a probabilistic representational model, the results are computed by averaging $K = 50$ samples from the learned distribution. Importantly, unlike the deterministic ResNet50-FCN, UQN can deliver the uncertainty map as meaningful guidance for our subsequent operations. Then, with our UGT, we observe clear performance improvements on all benchmarks. This is because UGT is able to reason over context to significantly enhance underlying representations. It should be noted that our UGT used here does not include the global semantic information from PT (*i.e.,* we remove the cross-attention module from UGT). Finally, with the aid of PT, our fully-equipped UGTR achieves the best performance. This indicates that semantic information is also important, because it can help the model infer local information reliably.

Superiority of the Prototyping Transformer. To under-

(a) Image (b) GT (c) UGTR(ours) (d) SINet (e) EGNet (f) PoolNet (g) PFANet

Figure 6: **Qualitative comparisons** between different models: (c) UGTR, (d) SINet [12], (e) EGNet [59], (f) POOLNet [33], and (g) PFANet [60]. Our UGTR can accurately identity the camouflaged object in different challenging scenarios.

Settings	CAMO-Test [29]				COD10K-Test [12]			
	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$
T=8,K=50	0.778	0.851	0.679	0.088	0.806	0.838	0.660	0.037
T=16,K=50	0.785	0.859	0.686	0.086	0.818	0.850	0.667	0.035
T=32,K=50	0.782	0.859	0.685	0.086	0.817	0.854	0.670	0.035
T=16,K=10	0.777	0.851	0.676	0.087	0.815	0.846	0.666	0.036
T=16,K=50	0.785	0.859	0.686	0.086	0.818	0.850	0.667	0.035
T=16,K=100	0.788	0.858	0.682	0.086	0.817	0.856	0.668	0.035

Table 5: **Analysis** of parameter settings on CAMO `test` and COD10K `test`. 'T' means the number of prototypes and 'K' means the number of extracted samples.

stand the role of our PT in learning high-level semantics, we conduct a comparative analysis on our prototyping approach. First, we replace PT with conventional strategies / modules: i) global average pooling (GAP) and ii) multi-scale pooling (MSP) [58]. As shown in Table 3, PT outperforms other strategies, while $\mathcal{L}_{\mathbf{PL}}$ brings additional performance boosts. Second, we show that $T = 16$ protoypes can ensure reliable performance in Table 5. Finally, we provide the visualizations of some prototypes in Figure 5. Clearly, our PT can group semantically similar pixels together, which helps to summarize / understand the given scene in a more global view.

Uncertainty-Guided Random Masking. UGRM is another important component in our UGTR. It works as a *hard-example-mining* module, which induces our UGT to focus on difficult (uncertain) regions during training. To show its comparative advantages, we provide two baseline models. The first one is trained by simply removing the UGRM in the training phase, and the second employs the widely-used random masking (RM) technique [50] for training. As shown in Table 4, our UGRM can bring clear improvements across different metrics. Thus, we can conclude that our UGRM can help the model learn to better deal with uncertain information and improve overall accuracy.

Do We Really Need Uncertainty? To answer this question, we first provide some visual examples in Figure 7. We observe that the estimated uncertainty maps consistently highlight object boundaries and indistinguishable textures, which means that those regions confuse deep COD model.

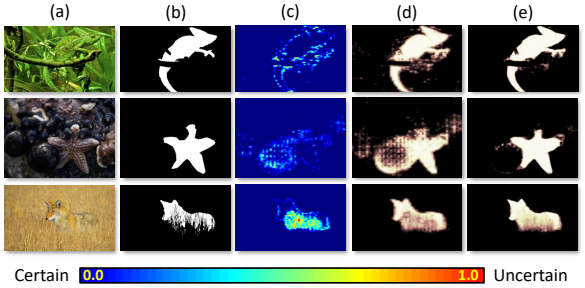

Figure 7: **Visualization of uncertainty map**. (a) Input images; (b) Ground truth; (c) Uncertainty map; (d) Initial prediction; (e) Final prediction by our approach.

Our observation is entirely consistent with the discoveries made by biological research [22, 52, 53] which find capturing the true body / object shape is the key to decamouflaging for human beings. Uncertainty quantification is meaningful because it **i**) increases the interpretability of COD models, **ii**) reveals the shortcomings of conventional solution and, **iii**) points out the future directions of COD research. UGT is trained to focus on uncertain regions so that significant accuracy improvement can be achieved by our UGTR. Moreover, we show that $K = 50$ samples from UNQ can quantify the uncertainty very well in Table 5.

5. Conclusion

In this paper, we inject Bayesian learning into Transformer based reasoning to handle camouflaged object detection. A probabilistic representational model (UQN) is learned to obtain initial estimates and associated uncertainties, and then transform-based modules (PT and UGT) further reason with context information to overcome ambiguity. All modules are carefully integrated as a new learning paradigm for COD, which mingles both probabilistic and deterministic procedures. We believe that our approach is also potentially applicable for other computer vision tasks.

Acknowledgements. This research was funded in part by the National Natural Science Foundation of China (U1964203) and the National Key R&D Program Project of China (2017YFB0102603).

References

[1] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *ICML*, 2015. 2, 4

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 3, 5

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

[4] Hung-Kuo Chu, Wei-Hsin Hsu, Niloy J Mitra, Daniel Cohen-Or, Tien-Tsin Wong, and Tong-Yee Lee. Camouflage images. *TOG*, 2010. 1

[5] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-detr: Unsupervised pre-training for object detection with transformers. In *CVPR*, 2021. 3

[6] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 1977. 5

[7] 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. In *ICLR*, 2021. 2

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 3, 5

[9] 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 *ECCV*, 2018. 1

[10] Deng-Ping Fan, Ming-Ming Cheng, Yun Liu, Tao Li, and Ali Borji. Structure-measure: A new way to evaluate foreground maps. In *ICCV*, 2017. 6, 7

[11] Deng-Ping Fan, Ge-Peng Ji, Xuebin Qin, and Ming-Ming Cheng. Cognitive vision inspired object segmentation metric and loss function. *SSI*, 2021. 6, 7

[12] Deng-Ping Fan, Ge-Peng Ji, Guolei Sun, Ming-Ming Cheng, Jianbing Shen, and Ling Shao. Camouflaged object detection. In *CVPR*, 2020. 1, 2, 3, 6, 7, 8

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

[14] Yarin Gal and Zoubin Ghahramani. Bayesian convolutional neural networks with bernoulli approximate variational inference. *arXiv preprint arXiv:1506.02158*, 2015. 2

[15] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, 2016. 2, 4

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 1, 6

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

[18] Jianqin Yin Yanbin Han Wendi Hou and Jinping Li. Detection of the mobile object with camouflage color under dynamic background based on optical flow. *Procedia Engineering*, 2011. 1, 2

[19] Po-Yu Huang, Wan-Ting Hsu, Chun-Yueh Chiu, Ting-Fan Wu, and Min Sun. Efficient uncertainty estimation for semantic segmentation in videos. In *ECCV*, 2018. 2

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

[21] Iván Huerta, Daniel Rowe, Mikhail Mozerov, and Jordi Gonzàlez. Improving background subtraction based on a casuistry of colour-motion segmentation problems. In *ICPRIA*, 2007. 1, 2

[22] Changku Kang, Martin Stevens, Jong-yeol Moon, Sang-Im Lee, and Piotr G Jablonski. Camouflage through behavior in moths: the role of background matching and disruptive coloration. *Behavioral Ecology*, 2015. 8

[23] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv preprint arXiv:1511.02680*, 2015. 2, 4

[24] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *NeurIPS*, 2017. 2, 3, 4

[25] Diederik P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. *arXiv preprint arXiv:1506.02557*, 2015. 2, 4

[26] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2, 4

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

[28] Yongchan Kwon, Joong-Ho Won, Beom Joon Kim, and Myunghee Cho Paik. Uncertainty quantification using bayesian neural networks in classification: Application to biomedical image segmentation. *Computational Statistics & Data Analysis*, 2020. 2

[29] Trung-Nghia Le, Tam V. Nguyen, Zhongliang Nie, Minh-Triet Tran, and Akihiro Sugimoto. Anabranch network for camouflaged object segmentation. *CVIU*, 2019. 1, 2, 3, 6, 7, 8

[30] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *CVPR*, 2021. 3

[31] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 1, 6

[32] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 1

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

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

[35] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016. 1

[36] Yunqiu Lv, Jing Zhang, Yuchao Dai, Li Aixuan, Bowen Liu, Nick Barnes, and Deng-Ping Fan. Simultaneously localize, segment and rank the camouflaged objects. In *CVPR*, 2021. 2, 3, 6

[37] Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. *NeurIPS*, 2019. 2

[38] Ran Margolin, Lihi Zelnik-Manor, and Ayellet Tal. How to evaluate foreground maps? In *CVPR*, 2014. 6, 7

[39] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. *arXiv preprint arXiv:2101.02702*, 2021. 3

[40] Yuxin Pan, Yiwang Chen, Qiang Fu, Ping Zhang, and Xin Xu. Study on the camouflaged target detection method based on 3d convexity. *Modern Applied Science*, 2011. 1, 2

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

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

[43] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 1

[44] Hippolyt Ritter, Aleksandar Botev, and David Barber. A scalable laplace approximation for neural networks. In *ICLR*, 2018. 2

[45] Sujit K Singh, Chitra A Dhawale, and Sanjay Misra. Survey of object detection methods in camouflaged image. *IERI Procedia*, 2013. 1

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

[47] Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. Bottleneck transformers for visual recognition. *arXiv preprint arXiv:2101.11605*, 2021. 3, 5

[48] Peize Sun, Yi Jiang, Rufeng Zhang, Enze Xie, Jinkun Cao, Xinting Hu, Tao Kong, Zehuan Yuan, Changhu Wang, and Ping Luo. Transtrack: Multiple-object tracking with transformer. *arXiv preprint arXiv:2012.15460*, 2020. 3

[49] Gerald Handerson Thayer. *Concealing-coloration in the animal kingdom: an exposition of the laws of disguise through color and pattern*. Macmillan Company, 1918. 1

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2, 3, 6, 7, 8

[51] Hao Wang and Dit-Yan Yeung. A survey on bayesian deep learning. *ACM Computing Surveys*, 2020. 2

[52] Richard J Webster. Does disruptive camouflage conceal edges and features? *Current Zoology*, 2015. 8

[53] Richard J Webster, Christopher Hassall, Chris M Herdman, Jean-Guy J Godin, and Thomas N Sherratt. Disruptive camouflage impairs object recognition. *Biology Letters*, 2013. 8

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

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

[56] Tianyuan Yu, Da Li, Yongxin Yang, Timothy M Hospedales, and Tao Xiang. Robust person re-identification by modelling feature uncertainty. In *ICCV*, 2019. 2, 3, 4

[57] Qiang Zhai, Xin Li, Fan Yang, Chenglizhao Chen, Hong Cheng, and Deng-Ping Fan. Mutual graph learning for camouflaged object detection. In *CVPR*, 2021. 1, 2, 3

[58] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 6, 7, 8

[59] Jia-Xing Zhao, Jiang-Jiang Liu, Deng-Ping Fan, Yang Cao, Jufeng Yang, and Ming-Ming Cheng. Egnet: Edge guidance network for salient object detection. In *ICCV*, 2019. 1, 6, 8

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

[61] Xiaoqi Zhao, Youwei Pang, Lihe Zhang, Huchuan Lu, and Lei Zhang. Suppress and balance: A simple gated network for salient object detection. In *ECCV*, 2020. 1

[62] Olga Zhaxybayeva and J Peter Gogarten. Bootstrap, bayesian probability and maximum likelihood mapping: exploring new tools for comparative genome analyses. *BMC genomics*, 2002. 2

[63] Minghang Zheng, Peng Gao, Xiaogang Wang, Hongsheng Li, and Hao Dong. End-to-end object detection with adaptive clustering transformer. *arXiv preprint arXiv:2011.09315*, 2020. 5

[64] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. *arXiv preprint arXiv:2012.15840*, 2020. 3

[65] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In *DLMIA*, pages 3–11, 2018. 6

[66] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. 2021. 3, 5