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 camouflaging difficult for both humans and machines.

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...
propose the **Uncertainty-Guided Transformer Reasoning** (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 outputting 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.

Inspired 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 $\{\mathbf{I}_i, \mathbf{C}_i\}_{i=1}^N$, our task is to learn a mapping function $F_\theta$ parameterized by weights $\Theta$ 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 $F_\theta$ 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 Uncertainty-Guided Transformer Reasoning model (UGTR). As shown in Figure 2, UGTR includes three major components:

- **Uncertainty Quantification Network** (§ 3.2). Our uncertainty quantification network (UQN), denoted as $F_{\theta_1}$, contains two parts: feature extractor (backbone) $F_{\theta_1}$ and probabilistic module $F_{\theta_2}$. For $F_{\theta_1}$, it takes the RGB image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ as input, and produces feature embeddings $\mathbf{F} \in \mathbb{R}^{h \times w \times c}$; $\mathbf{I} \xrightarrow{F_{\theta_1}} \mathbf{F}$. Then, $F_{\theta_2}$ 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 $\mathbf{\mu} \in \mathbb{R}^{h \times w \times 1}$ and variance map $\mathbf{\sigma} \in \mathbb{R}^{h \times w \times 1}$; $\mathbf{F} \xrightarrow{F_{\theta_2}} (\mathbf{\mu}, \mathbf{\sigma})$, and draw $K$ samples (camouflage maps) from the distributions to produce the uncertainty map.

- **Prototyping Transformer** (§ 3.3). Our prototyping transformer (PT) $F_{\delta}$ works as an auxiliary module to learn and reason over higher-level semantics. It transforms $\mathbf{F}$ to $t$ semantic prototypes: $\mathbf{F} \xrightarrow{F_{\delta}} \mathbf{X}$, 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
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.

X can be used to assist final reasoning.

- Uncertainty-Guided Transformer (§ 3.4). Finally, our uncertainty guided transformer (UGT) $F_\theta$ takes $F$, $X$ and $U$ as inputs, and produces the refined features $\tilde{F} \in \mathbb{R}^{(h \times w) \times c}$ for COD in a sequence-to-sequence manner:

$$(F, X, U) \xrightarrow{F_\theta} \tilde{F}.$$ 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 $U$ 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 $\tilde{F}$ so that the entire decamouflaging process involves both probabilistic and deterministic procedures. To achieve the final predictions, $\tilde{F}$ can be easily mapped to $C$ with a readout function $F_{\text{out}}$: $\tilde{F} \xrightarrow{F_{\text{out}}} C$, which is composed of a reshape layer, a $1 \times 1$ 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) $F_\theta$ as a probabilistic representational model to measure uncertainty. Therefore, in this stage, what $F_\theta$ delivers for each pixel (e.g. the pixel $p$) is a distribution parameterized by mean $\mu_p$ and variance $\sigma_p$ 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: $c_p \sim \mathcal{N}(\mu_p, \sigma_p)$, where $\mu_p$ and $\sigma_p$ are produced by UQN. As already observed by existing works [15,23,24,56], when we use random sample to train $F_\theta$, 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 $\epsilon_p$ from the standard Gaussian distribution $\mathcal{N}(0, I)$ randomly, i.e., $c_p \sim \mathcal{N}(0, I)$, and then obtain the sample by computing $\mu_p + \epsilon_p \sigma_p$. By doing so, the gradients can be propagated backward to optimize the parameters $\theta$.

#### 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 $\mu$ and $\sigma$, respectively. As shown in Figure 3, our UQN includes two major parts: backbone $F_{\theta_1}$ and probabilistic module $F_{\theta_2}$. Concretely, given the input image $I \in \mathbb{R}^{H \times W \times 3}$, the backbone $F_{\theta_1}$ is employed to obtain c-dimensional feature embeddings: $F = F_{\theta_1}(I) \in \mathbb{R}^{h \times w \times c}$. Then, a two-branch probabilistic module $F_{\theta_2}$ further transfers $F$ to $\mu_p$ and $\sigma_p$:

$$\mu = F_{\theta_2}(F), \quad \sigma = F_{\theta_2}(F),$$

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

#### Distribution Learning Loss $L_{\text{DLL}}$

To train our UQN, we design the following loss function $L_{\text{DLL}}$, which is a weighted combination of a standard BCE loss and a Kullback-Leibler (KL) divergence [26]:

$$L_{\text{DLL}} = L_{\text{BCE}}(c^{(k)}(\text{gt})) + \eta \cdot D(\mathcal{N}(\mu, \sigma) \parallel \mathcal{N}(0, I))$$

where $\eta$ 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 $C_{\text{gt}}$ denotes the ground truth.

#### Uncertainty Quantification

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

$$U = \text{Norm}(\text{Var}(C_{\text{init}})),$$

where $U \in \mathbb{R}^{h \times w \times 1}$ means the uncertainty map, $\text{Norm}(\cdot)$ is the mean-max normalization operation and $\text{Var}(\cdot)$ 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,

\footnote{To promote diversity, we only draw one example to compute the loss.}
The two operations above (Eq. 4 and Eq. 5) will be repeated for multiple times to find the cluster centroids as \( \mathbf{X}_{\text{init}} \). 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 \( \mathbf{X}_{\text{init}} \) 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:

\[
Q_i = \mathbf{X}_{\text{init}}W^Q_i, ~ K_i = \mathbf{X}_{\text{init}}W^K_i, ~ V_i = \mathbf{X}_{\text{init}}W^V_i, \\
\text{Head}_i = \text{Attention}(Q_i, K_i, V_i), \\
MH(Q, K, V) = \text{Concat}(\text{Head}_1, \ldots, \text{Head}_M),
\]

where \( W^Q_i, W^K_i \) and \( W^V_i \) 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 position features \( \mathbf{P}_X \) by:

\[
p_{x,t} = \frac{\sum_{p=1}^{hw} A_{p,t} f_p}{\sum_{p=1}^{hw} A_{p,t}}.
\]

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 \( \mathcal{L}_{PL} \).** 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 \( \mathcal{L}_{PL} \):

\[
\mathcal{L}_{PL} = \sum_{x_i, x_j \in \mathbf{X}} \max((m - \text{dist}(x_i, x_j)), 0),
\]

where \( m \) is a pre-set threshold. \( \mathcal{L}_{PL} \) 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 \( \tilde{\mathbf{F}} \). Formally, our

\[ ^2 \text{Each transformer block consists of a multi-head self-attention module and a feed forward network (FFN).} \]
UGT $\mathcal{F}_\phi$ takes $F \in \mathbb{R}^{c \times hw}$, $U \in \mathbb{R}^{1 \times hw}$ and $X \in \mathbb{R}^{c \times T}$ as inputs and produces $\tilde{F} \in \mathbb{R}^{c \times hw}$ as output. The main difference from PT is that we feed the uncertainty-aware features $F_U = F \otimes (1 - U)$ to transformers ($\otimes$ element-wise multiply) and add a cross-attention layer in each transformer block so that $X$ can be incorporated for representation learning:

$$
\tilde{Q}_i = F_U W^Q_i, \tilde{K}_i = X W^K_i, \tilde{V}_i = X W^V_i, \\
\text{Head}_i = \text{Attention}(\tilde{Q}_i, \tilde{K}_i, \tilde{V}_i), \quad (9) \\
M H(Q, K, V) = \text{Concat}(\text{Head}_1, \ldots, \text{Head}_M),
$$

where $\tilde{Q}_i$, $\tilde{K}_i$ and $\tilde{V}_i$ 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 $U$. Formally, let us denote $R \in \mathbb{R}^{1 \times hw}$ as a random probability map, where $r_p \in [0, 1]$. We mask out those features $f_p \in F_U$ 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 ($\tilde{F}$) is mapped to final predictions by using a simple readout function $F_{\text{out}}$, which is composed of a reshape layer, a $1 \times 1$ convolutional layer and upsampling layer. The loss function for our fully-equipped UGTR is a combination of multiple loss functions:

$$
\mathcal{L}_{\text{UGTR}} = \mathcal{L}_{\text{DDL}} + \eta \mathcal{L}_{\text{PT}} + \omega \mathcal{L}_{\text{BCE}} \quad (10)
$$

where $\eta$ and $\omega$ mean the combination weights, and $\mathcal{L}_{\text{BCE}}$ 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 ($E_\phi$) [11], mean S-measure ($S_\alpha$) [10] and F-measure ($F^\beta_\gamma$) [38].

**Training Settings.** During training, the underlying backbone $F_{\text{B}}$ is initialized by ResNet-50 [17] pre-trained on ImageNet [27], and the remaining modules (i.e., probabilistic module $F_{\text{B}}$, prototyping transformer $F_{\phi}$ and uncertainty-guided transformer $F_{\phi}$) 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: $lr = \text{base}\cdot\text{lr} \cdot (1 - \frac{t}{\text{lr}_\text{max}})^{\text{power}}$. We empirically set the base learning rate $\text{base}\cdot\text{lr}$ to $10^{-7}$ and $\text{power}$ 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...
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].

**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 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-
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.

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


[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


[18] Jiaojin 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


[21] Iván Huerta, Daniel Rowe, Mikhail Mozorev, and Jordi González. Improving background subtraction based on a causality of colour-motion segmentation problems. In ICPRIA, 2019. 1, 2


[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
[38] Ran Margolin, Lihi Zelnik-Manor, and Ayellet Tal. How to evaluate foreground maps? In CVPR, 2014. 6, 7
[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
[42] Xuebin Qin, Zchen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jagersand. Basnet: Boundary-aware salient object detection. In CVPR. 2019. 6
[57] Zhaibai Zhao, 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, Xiaoang 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