Modeling the Distributional Uncertainty for Salient Object Detection Models

Xinyu Tian[1] Jing Zhang[2]✉ Mochu Xiang[1] Yuchao Dai[1]✉
[1] Northwestern Polytechnical University, China [2] Australian National University, Australia

Abstract

Most of the existing salient object detection (SOD) models focus on improving the overall model performance, without explicitly explaining the discrepancy between the training and testing distributions. In this paper, we investigate a particular type of epistemic uncertainty, namely distributional uncertainty, for salient object detection. Specifically, for the first time, we explore the existing class-aware distribution gap exploration techniques, i.e. *long-tail learning, single-model uncertainty modeling and test-time strategies, and adapt them to model the distributional uncertainty for our class-agnostic task. We define test sample that is dissimilar to the training dataset as being "out-of-distribution" (OOD) samples. Different from the conventional OOD definition, where OOD samples are those not belonging to the closed-world training categories, OOD samples for SOD are those break the basic priors of saliency,* i.e. *center prior, color contrast prior, compactness prior and* etc.*, indicating OOD as being "continuous" instead of being discrete for our task. We've carried out extensive experimental results to verify effectiveness of existing distribution gap modeling techniques for SOD, and conclude that both train-time single-model uncertainty estimation techniques and weight-regularization solutions that preventing model activation from drifting too much are promising directions for modeling distributional uncertainty for SOD.*

1. Introduction

Saliency detection (or salient object detection, SOD) [6, 11, 14, 40, 44, 64, 65, 67–69, 74–76, 78] aims to localize object(s) that attract human attention. Most of the existing techniques focus on improving model performance on benchmark testing dataset without explicitly explaining the distribution gap issue within this task. In this paper, we aim to explore the distributional uncertainty for better understanding of the trained saliency detection models.

Jing Zhang (zjnwpu@gmail.com) and Yuchao Dai (daiyuchao@nwpu.edu.cn) are the corresponding authors.
Our code is publicly available at: https://npucvr.github.io/Distributional_uncer/.

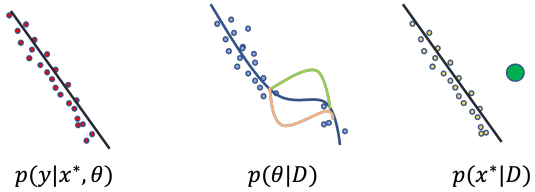

Figure 1. Visualization of different types of uncertainty, where aleatoric uncertainty ($p(y|x^*,\theta)$) is caused by the inherent randomness of the data, model uncertainty ($p(\theta|D)$) happens when there exists low-density region, leading to multiple solutions within this region, and distributional uncertainty ($p(x^*|D)$) occurs when the test sample x^* fails to fit in the model based on the training dataset D.

Background: Suppose the training dataset is $D = \{x_i, y_i\}_{i=1}^N$ of size N is sampled from a joint data distribution $p(x, y)$, where i indexes the samples and is omitted when it's clear. The conventional classifier is trained to maximize the conditional log-likelihood $\log p_\theta(y|x)$, where θ represents model parameters. When deploying the trained model in real-world, its performance depends on whether the test sample x^* is from the same joint data distribution $p(x, y)$. For x^* from $p(x, y)$ (indicating x^* is in-distribution sample), its performance is impressive. However, when it's from a different distribution other than $p(x, y)$ (*i.e.* x^* is out-of-distribution sample), the resulting $p(y|x^*)$ often yield incorrect predictions with high confidence. The main reason is that $p(y|x^*)$ does not fit a probability distribution over the whole joint data space. To fix the above issue, deep hybrid models (DHMs) [5, 18, 46, 73] can be used to model the joint distribution: $p(x, y) = p(y|x)p(x)$. Although the trained model may still inaccurately assign high confidence $p(y|x^*)$ for out-of-distribution sample x^*, effective marginal density modeling of $p(x^*)$ can produce low density for it, leading to reliable $p(x^*, y)$.

Given a testing sample x^*, with a deep hybrid model, [5] proposes to factorize the posterior joint distribution $p(x^*, y|\theta, D)$ via:

$$p(x^*, y|\theta, D) = \underbrace{p(y|x^*, \theta)}_{\text{data}} \underbrace{p(x^*|D)}_{\text{distributional}} \underbrace{p(\theta|D)}_{\text{model}}, \quad (1)$$

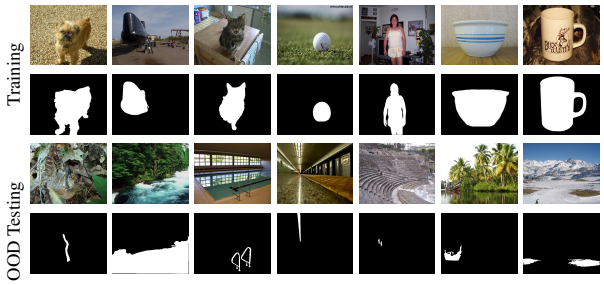

Figure 2. "OOD" samples for salient object detection. Different from the class-aware tasks, OOD for saliency detection is continuous, which can be defined as attributes that break the basic saliency priors, *i.e*. center prior, contrast prior, compactness prior, *etc*.

where the basic assumption is that $p(x^*|\theta, D) = p(x^*|D)$, which is true as θ is obtained based on D. $p(y|x^*, \theta)$ in Eq. 1 is used to model data uncertainty or aleatoric uncertainty [31], which is inherent in the data generation process. $p(x^*|D)$ represents distributional uncertainty, explaining the distribution gap between the test sample and the training dataset. $p(\theta|D)$ is model uncertainty, representing the uncertainty of model parameters given the current training dataset (see Fig. 1). The latter two can be combined and termed as "epistemic uncertainty" [31].

As the aleatoric uncertainty captures the inherent stochastic, it usually cannot be explained away with more data. On the contrary, more training data can reduce the model uncertainty. We claim that "distributional uncertainty", indicating the degree of "out-of-distribution" (OOD), is harder to reduce than "model uncertainty". Although more diverse data can reduce the "distributional uncertainty" to some extend, as we cannot estimate an unbiased testing distribution, distributional uncertainty cannot be completely explained away. In this paper, we focus on modeling the distributional uncertainty for saliency detection, and define OOD samples for saliency detection as those break the basic saliency priors, *i.e*. center prior, color contrast prior, compactness prior and *etc*. (see Fig. 2). In this case, saliency OOD is "continuous" instead of discrete.

For the first time, we aim to explore distributional uncertainty estimation for saliency detection. Specifically, we investigate the existing class-aware distribution gap exploration techniques and adapt them to model the distributional uncertainty for our class-agnostic task (sec. 2). We also perform extensive experiments in sec. 3 to explain both the advantages and limitations of each technique for our task. Based on the extensive experiments, we conclude: **1)** With the ensemble structure, Deep ensemble [34] produces more accurate predictions with high calibration degree [19] compared with Monte Carlo (MC) dropout [15] for salient object detection; **2)** The categorical distribution based long-tail solutions [10, 13, 29, 37, 56, 57, 59, 62, 63] fail to generalize well to the continuous saliency detection task; **3)** The single-model uncertainty methods [9, 22, 25] are effective in exploring distributional uncertainty for SOD, especially the training-time based techniques [9, 54]; **4)** For pixel-level prediction tasks, the test-time training methods [3, 4, 7, 12, 16, 20, 51, 55, 61] are prone to generate too confidence predictions in the uncertain area. Based on data augmentation, the test-time testing solutions [8, 32, 33, 38, 42, 43, 49, 52, 53] show potential in producing reliable distributional uncertainty for salient object detection.

2. Distributional Uncertainty Modeling

As shown in Eq. 1, distributional uncertainty indicates the gap between the test sample x^* and the training dataset D. We mainly explore three types of distribution gap modeling techniques [13, 22, 45, 49, 66], including long-tail learning techniques (sec. 2.1), single model uncertainty estimation (sec. 2.2) and test-time strategies (sec. 2.3). We will introduce the state-of-the-art solutions of each direction, and apply them to our class-agnostic binary segmentation task for distributional uncertainty estimation.

Setup: For easier presentation, we explain the experiments setup first. We take ResNet50 [21] as our backbone model $f_\theta = \{s_k\}_{k=1}^4$, where channel sizes of the backbone features are 256, 512, 1024 and 2048 respectively. To relief the huge memory requirement and also obtain larger receptive field, we feed the backbone features to four different multi-scale dilated convolutional blocks [72] with Batch Renormalization layers [27] and obtain new backbone features $f_\theta^b = \{s'_k\}_{k=1}^4$ of the same channel size as 32, and we define f^b as the encoder of our framework. We then feed $\{s'_k\}_{k=1}^4$ to decoder f^d from [50] to generate the saliency prediction $f_\theta^d(f_\theta^b(x))$ for input RGB image x.

2.1. Long-tail learning to overcome distribution bias

Long-tail learning based methods [47] are designed to address the problem of class imbalance in the training data. In this case, the "distribution gap" can be explained as the gap between long-tailed training and uniform testing. Existing long-tail solutions can be roughly divided into data resampling based methods [10, 13, 29, 62] and loss reweighting strategies [37, 56, 57, 59, 63]. The former over-samples the training data from tail classes and under-samples those from the head classes to achieve data re-balance. The latter regularizes model parameters to pay more attention to the tail classes via class-balanced loss function.

Model-rebalance techniques: As long-tailed training leads to biased model, model-rebalance techniques [77] aim to debiase the model directly. Among them, [77] designs a diverse expert learning network to model different distributions, and employs a test-time self-supervised learning method to learn the weights for aggregation of diverse experts. Specifically, the network has a shared feature ex-

tractor and three category prediction heads that simulate long-tailed distribution, uniform distribution, and inverse long-tailed distribution, respectively. The final loss function is then defined based on outputs from the three prediction heads. Further, [77] performs test-time self-supervised learning to achieve prediction consistency, and learn the aggregation weights $\{\alpha_l\}_{l=1}^3$ to generate the final prediction.

Loss reweighting techniques: Model-rebalance techniques [77] achieve debiased models by assuming various test distributions. Alternatively, loss-reweighting methods intend to balance the contributions of the head classes and tail classes. Among them, [2] introduces a two-stage learning method, which first learns feature representation via weight decay, and then uses the combination of class-balanced loss function $\mathcal{L}_{CB}$, weight decay and MaxNorm [23] to learn the weights of the classification layer, where the class-balanced loss function can be written as:

$$\mathcal{L}_{CB}(x) = \frac{1-\beta}{1-\beta^{N_y}}\log(\hat{y}), \tag{2}$$

where $\{N_c\}_{c=1}^C$ is the number of class c in training dataset, y is the ground truth class label, $\hat{y}$ is the predicted class probability, and β is a hyperparameter.

Post-hoc techniques: Instead of training the model with model/loss rebalance strategies, long-tail learning can also be achieved via post-hoc techniques, *i.e*. [48] (NorCal) optimizes the general Softmax process by counting the number of each class $\{N_c\}_{c=1}^C$ and calculating the class weights. The probability that the image x is predicted to be class c is then defined as:

$$p(c|x) = \frac{\exp(\phi_c(x))/\alpha_c}{\sum_{c'=1}^{C}\exp(\phi'_c(x))/\alpha'_c + \exp(\phi_{c+1}(x))}, \tag{3}$$

where α_c is monotonically increasing with respect to N_c, ϕ_c is the predicted logit of class c and $c+1$ is the background class for object detection task. In this way, the scores for head classes will be suppressed.

Long-tail techniques for saliency detection: The main challenge of applying existing long-tail learning methods to saliency detection is that tail "categories" of class-aware tasks, *e.g*. object detection, semantic segmentation, image classification, are easy to define, while it's hard to identify tail classes for our class-agnostic task, as saliency is "attribute" based, which is continuous. Following the conventional long-tail learning techniques, we introduce "continuous version" long-tail learning for saliency detection.

1) Model-rebalance based saliency distributional uncertainty modeling: To adapt [77] for saliency detection, we construct a multi-head dense prediction network with a shared encoder f_θ^b and three decoders $\{f_{\theta_l}^d\}_{l=1}^3$ that simulate the three different distributions, *i.e*. long-tailed distribution, uniform distribution, and inverse long-tailed distribution with different loss function. We first learn this multi-head model where the loss function is defined as sum of these three distribution-head losses. As a binary dense prediction task, the outputs $\hat{y}_l \in \mathbb{R}^{C\times H\times W}$ of these three heads are pixel-wise predictions after Softmax with $C=2$.

After the first stage training of the multi-head model, the test-time training is performed to learn the aggregation weights $\{\alpha_l\}_{l=1}^3$ for each test sample x_t with loss function defined as:

$$\mathcal{L}_\alpha(x_t) = -\text{sim}(\hat{y}(x_t), \hat{y}'(x'_t)), \tag{4}$$

where $\hat{y}(x_t) = \sum_{l=1}^3 \alpha_l \cdot \hat{y}_l(x_t)$ is the final prediction of x_t, $\text{sim}(,)$ is the cosine similarity, x'_t represents the augmented image, where we use the horizontal flip and $\hat{y}'$ represents the inverse augmentation operation on the output.

2) Loss-reweighting based saliency detection: As a two-stage training framework, we perform loss-reweighting based saliency detection following [2], where within the first stage training, we use conventional models with weight decay to learn feature representation. Then for the second stage training, class weighted loss function is used to balance the class distribution via:

$$\mathcal{L}_{CB}(x_u) = \alpha_u(y_u\text{log}(\hat{y}_u) + (1-y_u)\text{log}(1-\hat{y}_u)), \tag{5}$$

where $\alpha_u = (1-\beta)/(1-\beta^{N_{y_u}})$ is a pixel-wise class-balanced wight, and y_u is the ground truth class label of pixel u. In practice, for the second stage training, we freeze all parameters from the first stage training except the last prediction convolutional layer and use weight decay, MaxNorm [23] and pixel-wise class-balanced loss functions simultaneously to optimize the predictions.

3) Post-hoc long-tail learning for saliency detection: Based on our trained base model, we can use the class weighting function to recalibrate the output logit as a post-hoc technique to get the predicted probability. The probability that pixel x_u is predicted to be foreground is:

$$p_f(x_u) = \frac{\exp(\phi_f(x_u))/\alpha_f}{\exp(\phi_f(x_u))/\alpha_f + \exp(\phi_b(x_u))/\alpha_b}, \tag{6}$$

where α_f and α_b monotonically increase with respect to the number of foreground pixels N_f and background pixels N_b respectively. ϕ_f, ϕ_b are foreground and background logits respectively.

2.2. Single-model uncertainty

Uncertainty estimation aims to estimate uncertainty of the data or the model. As explained in Fig. 1, aleatoric uncertainty is inherent, which can not be explained away with more data. We focus on epistemic uncertainty, especially uncertainty to model the out-of-distribution samples, which shares the same idea as distributional uncertainty estimation. The typical solution to achieve out-of-distribution (OOD) detection is designing a OOD detector $g(x)$ based

on a score function confidence $h(x)$ to decide the input sample as in-distribution (ID) or out-of-distribution (OOD) via:

$$g(x) = \begin{cases} \text{ID}, & \text{if} \quad h(x) \geq \tau, \\ \text{OOD}, & \text{if} \quad h(x) < \tau, \end{cases} \tag{7}$$

where τ the OOD threshold, which is usually chosen so that a high fraction of ID data is correctly classified. The main focus of OOD detection is then to define reliable score $h(x)$, which can be achieved via post-hoc techniques, *i.e*. gradient based method [25], energy-score based method [41], training techniques, *i.e*. loss function regularization [9], data reprocessing [58], or feature regularization [54].

1) Post-hoc techniques: The post-hoc techniques either compute confidence directly from model output (maximum class probability(MCP) [22]) or re-interpret model output [17] from Softmax via an energy-based model formulation [35]. There also exists solutions that renormalize the logit space based on statistics from the training dataset [28] or compute directly the Kullback–Leibler divergence of output distribution from the uniform distribution [36], where the basic assumption is the OOD samples should have uniform predictive distribution across the categories.

The basic assumption of MCP [22] is that correctly classified samples tend to have higher maximum softmax probabilities than the erroneously classified ones or the out-of-distribution ones. In this case, [22] defines the maximum softmax probability as confidence score. Specifically, for pixel u, it's MCP (confidence) is obtained via $h_u = \max\{p_f, p_b\}$, where p_f and p_b are the probability of pixel u is predicted to foreground/background for our binary segmentation task.

Based on the inherent connection between energy-based model [35] and the Softmax function within the modern machine learning framework, [41] introduces energy score. $E_\theta(x)$ is the free energy, which is defined as: $E_\theta(x) = -\log \int_{y'} \exp^{-E_\theta(x,y')} dy'$. Similarly, given the classification setting with $E_\theta(x, y = c) = -f_\theta(x)_c$, we have $E_\theta(x) = -\log \sum_{c=1}^{C} \exp(f_\theta(x)_c)$. [41] defines confidence score as negative free energy, leading to $h = -E_\theta(x)$.

SML [28] calculates the mean and variance of the foreground (m_f, v_f) and background logits (m_b, v_b) on the training set, which are used to standardize the max logits. The max logit of pixel u is $\phi_m = \max(\phi_{u,f}, \phi_{u,b})$, and the standardized max logit is $\phi_s = (\phi_m - m_f)/v_f$ if the label of pixel u is foreground. Boundary suppression and smoothing operations are then used to eliminate some false positives and get the confidence map.

Gradient based Confidence (GC) [1, 24, 25, 36, 39] assumes uniform predictive distribution for the OOD samples. As an extension, ExGrad [26] calculates the confidence of pixel u as $h(u) = 1 - \sum_{c=1}^{C} p_c \cdot (1 - p_c) = 1 - 2 \cdot p_f \cdot p_b$.

2) Training techniques: Training a single-model uncertainty estimation network can be achieved via loss function regularization [9], data re-processing [58], or weight regularization [54]. [9] uses an additional CondifNet to learn the true class probability (TCP), which directly reflects the confidence of predictions. Higher TCP values response network predictions are of higher quality. To adapt it for our saliency detection task, we add an additional decoder on the base model to predict the confidence map $h \in \mathbb{R}^{H \times W}$, and its output is constrained by the MSE loss with the true class probability value of the prediction decoder:

$$\mathcal{L}^{tcp}(x_u) = (h_u - p_u^\star)^2, \tag{8}$$

where the $p_u^\star$ is the probability of pixel u is predicted as the ground truth class. The uncertainty is defined as $u = 1 - h$.

Alternatively, data-renormalization [58] can be used to achieve single model uncertainty estimation. [58] adopts prediction variation as model uncertainty by training on a dataset shifted by random constant biases. Specifically, the conventional model is to map the input x to label y directly, but [58] makes the model to learn the mapping from $(z, x - z)$ to y by setting a random anchor z. By changing the random anchor z, the model can generate multiple predictions about x, resulting in the model uncertainty.

In our experiment, we select a random image from the mini-batch as a random anchor z, concatenate the random anchor z with the residual between the input image and the anchor $x - z$ as the input to the model f_θ, and add an additional convolution layer at the beginning of the model to reduce the dimension to 3 in order to fit the pretrained weights. The output of the model is supervised by the saliency map corresponding to the input image x. The loss function of the model is:

$$\mathcal{L}^{dc}(x) = \mathcal{L}(f_\theta(z, x - z), y). \tag{9}$$

Instead of loss or data regularization, [54] (ReAct) performs rectified activation on the features $r(x)$ of the model through a set threshold α, and then feeds it to the later layer f_l to get the prediction. The process can be described as

$$\begin{aligned} r'(x) &= min(r(x), \alpha), \\ \hat{y} &= f_l(r'(x)), \end{aligned} \tag{10}$$

where the threshold α is set based on the p-th percentile of activations estimated on the in-distribution data, which can sufficiently preserve the activations for in-distribution data and truncate high activation to limit the effect of noise. We perform ReAct on the features of the trained base model penultimate layer, from which we statistically obtain a threshold α such that 90% of the activation values are less than α. After several counts, the threshold is set to 5. The features after ReAct are then fed into the last layer to obtain the predictions.

2.3. Test-time Strategies

Test-time strategies [8, 32, 33, 38, 42, 43, 49, 52, 53], including both test-time training and test-time augmentation. The motivation of test-time training(TTT) [3, 4, 7, 12, 16, 20, 51, 55, 61] is to adapt each testing sample to the trained model by optimizing its specific part of parameters, which is also defined as a one-sample learning problem. The loss function is mostly defined as the consistency of the prediction results of the data before and after the augmentation t:

$$\begin{aligned} \mathcal{L}^{TTT}(x) &= \mathcal{L}(f_\theta(x), f_\theta(t(x))), \\ \theta &\leftarrow \theta - r\nabla_\theta \mathcal{L}^{TTT}(x), \end{aligned} \tag{11}$$

where r is the learning rate.

Test-time augmentation (TTA) aims to improve network performance through data augmentation at test time without adding additional network parameters for training. Most test-time augmentation methods select multiple suitable augmentations $\mathcal{T}$ for test data from a large transformation candidates using some strategies, and integrate the augmented predictions to obtain the final prediction result, which can be written as:

$$y_{TTA}(x) = \sum_{t \in \mathcal{T}} \alpha_t \cdot f_\theta(t(x)), \tag{12}$$

where α_t is the integration weight corresponding to the transformation t.

Test-time strategies for saliency detection: [61] makes the trained model gradually adapt to the test data by adopting a self-supervised learning method. Specifically, [61] adopts a teacher-student network in test dataset, and we initialize both teacher and student network by our trained base model. The student network f_{θ_s} takes the original image as the input, and the teacher network f_{θ_t} takes the augmented image as the input. Here we use an image enhancement strategy t that adds random Gaussian noise, since inappropriate color/brightness changes may lead to saliency changes. The student network updates the parameters through the consistency loss of the teacher-student network outputs, and the teacher network uses the exponential moving average of the student network to update its parameters:

$$\begin{aligned} \mathcal{L}(x) &= \mathcal{L}(f_{\theta_s}(x), f_{\theta_t}(t(x))), \\ \theta_s &\leftarrow \theta_s - r\nabla_{\theta_s}\mathcal{L}(x), \\ \theta_t &\leftarrow \alpha\theta_t + (1-\alpha)\theta_s. \end{aligned} \tag{13}$$

To prevent the catastrophic forgetting of the student network, the parameters of the student network are randomly restored to the initial weights. However, due to the influence of the performance of the trained model, self-supervised learning on the teacher-student network may make the network focus on the results of mis-classified regions, resulting in performance degradation.

To avoid drifting the trained model too much, [8] selects the most suitable augmentations for the image by cycling. Firstly, the model is trained with a Spearman-aware ranking loss with a loss prediction network f_l that takes images as input to predict the relative magnitude of loss values, which is supervised by the actual loss between the outputs of prediction head and the ground-truth maps:

$$\mathcal{L}_{loss}(x) = \mathcal{L}_{rank}(f_l(x), \mathcal{L}(f_\theta(x), y)). \tag{14}$$

During testing, the loss values of a series of augmented images are first calculated by the loss prediction network, from which the augmentation t with the smallest loss is selected to transform the image, and then the augmented image is used as input to repeat the above process, which finally stops at a set number of times.

$$t = \min_{t \in \mathcal{T}} f_l(t(x)), x \leftarrow t(x), \tag{15}$$

such that the augmented image can be obtained after a series of selected augmentations, and is fed to the trained base model to obtain the prediction. Alternatively, the above cycle can be repeated several times to obtain multiple predictions, and the entropy values of the predictions are used to weight the multiple predictions and obtain the final prediction, making it possible to focus more on the prediction with low uncertainty. Since dense saliency prediction tasks require network prediction to correspond pixel-by-pixel to the ground-truth saliency map, it is difficult to apply some transformations such as clipping in TTA that will lose some pixels, and we adopt transformations by adding random Gaussian noise, horizontal flipping and size scaling to compose the transformation candidates.

3. Experiments

Training/testing dataset: Following the conventional training and testing settings, we train our models with the DUTS training dataset [60] of size $N = 10,553$. We then test on three benchmark testing datasets, including DUTS testing dataset [60], ECSSD [70] and DUT [71] dataset.

Evaluation metrics: We report model performance using three standard metrics, including maximum F-measure, IoU and Accuracy. F-measure is defined as: $F_\beta = \frac{(1+\beta^2)\text{Precision}\times\text{Recall}}{\beta^2\text{Precision}+\text{Recall}}$, where both Precision and Recall are based on binarized saliency prediction (with 256 uniformly distributed binarazation thresholds in the range of [0,255]) and the ground truth map. "$\max F_\beta$" (maximum F-measure) is reported as the maximum F_β. IoU (Intersection over Union) score is computed based on the binary predicted mask with adaptive thresholds and the ground-truth. Accuracy measures the proportion of pixels after binarization that have been correctly assigned to the salient foreground or background, where the binarization is also

Table 1. Performance of classic distribution bias modeling strategies.

Method	DUTS [60]					ECSSD [70]					DUT [71]				
	max F_β ↑	IoU ↑	Accuracy ↑	FPR95 ↓	AUROC ↑	max F_β ↑	IoU ↑	Accuracy ↑	FPR95 ↓	AUROC ↑	max F_β ↑	IoU ↑	Accuracy ↑	FPR95 ↓	AUROC ↑
Base	0.849	0.774	**0.965**	0.151	0.952	0.924	0.869	**0.968**	0.127	0.959	0.753	0.682	**0.946**	0.211	0.925
MCDropout [15]	0.846	0.772	0.964	0.150	0.952	0.924	0.871	**0.968**	0.126	0.959	0.750	0.680	0.945	0.211	0.924
DeepEnsemble [34]	**0.851**	**0.781**	**0.965**	**0.141**	**0.954**	**0.925**	**0.873**	**0.968**	**0.113**	**0.962**	**0.758**	**0.690**	0.945	**0.196**	**0.928**

Table 2. Performance of long-tail learning based distribution bias modeling strategies.

Method	DUTS [60]					ECSSD [70]					DUT [71]				
	max F_β ↑	IoU ↑	Accuracy ↑	FPR95 ↓	AUROC ↑	max F_β ↑	IoU ↑	Accuracy ↑	FPR95 ↓	AUROC ↑	max F_β ↑	IoU ↑	Accuracy ↑	FPR95 ↓	AUROC ↑
Base	**0.849**	**0.774**	**0.965**	**0.151**	**0.952**	**0.924**	**0.869**	**0.968**	**0.127**	**0.959**	**0.753**	**0.682**	**0.946**	**0.211**	**0.925**
TALT [77]	0.828	0.768	0.962	0.284	0.918	0.913	0.868	0.967	0.199	0.940	0.741	**0.682**	0.942	0.376	0.873
NorCal [48]	0.839	0.771	0.964	0.187	0.942	0.918	0.868	0.967	0.149	0.954	0.747	0.681	0.944	0.262	0.910
WB [2]	0.843	0.773	0.964	0.165	0.948	0.918	0.866	0.967	0.136	0.956	0.748	0.681	0.945	0.232	0.919

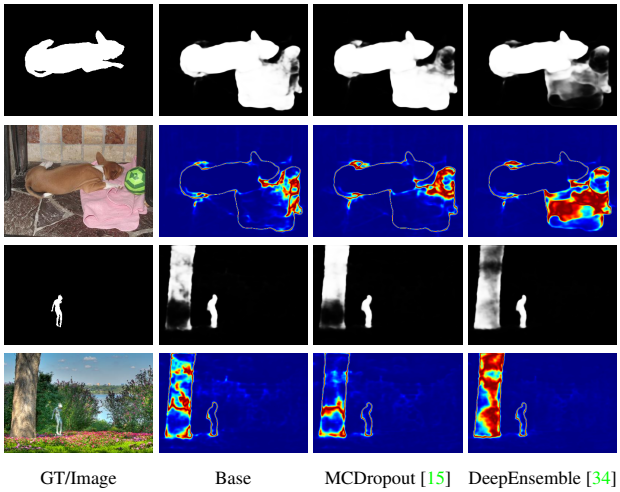

Figure 3. Visual comparison of [15] with [34]. The first column shows the input image and segmentation GT, and the other columns show the generated segmentation predictions and uncertainty maps. Compared to other models, the DeepEnsemble method integrates the information of multiple decoders, making it more accurate in distributional uncertainty modeling.

achieved via adaptive thresholding. To evaluate the distributional uncertainty modeling degree of the trained model, following metrics for out-of-distribution, we also report performance with area under receiver operating characteristics (AUROC) and false positive rate at a true positive rate of 95% (FPR95) since the rate of false positives in high- recall areas is crucial for safety-critical applications.

3.1. Evaluate Sample Difficulty

Due to the inherent "continuous" nature of saliency, we cannot directly prepare OOD samples based solely on the categories or distribution of categories. Alternatively, we adopt sample-difficulty indicator from [1] to roughly decide the difficulty of each test sample.

[1] measures sample difficulty by the gradient variance in the training phase. Specifically, we set multiple (K) checkpoints $\{f_k\}_{k=1}^K$ during training to obtain the gradient at the input $g_k \in \mathbb{R}^{3\times H\times W}$ for each sample. After training, we can obtain K gradient maps $\{g_k\}_{k=1}^K$ for each sample, and by calculating the variance, we can get the variance of gradient (VoG) score S_{VoG} via:

$$S_{VoG} = \frac{1}{HW}\sum_u \left(\frac{\sum_{k=1}^K (g_k(u)-\mu(u))^2}{\sqrt{K}}\right), \quad \mu = \frac{1}{K}\sum_{k=1}^K g_k, \tag{16}$$

where $\sum_u$ calculates pixel-wise sum across pixel u. Images with large VoG scores represent large gradient changes during the training phase and are identified as difficult samples. We define them as OOD samples in this paper (see Fig. 2). Unless stated otherwise, samples for the the visual comparisons are from the hard sample pool based on the VoG score.

3.2. Uncertainty Computation

Except for the methods that can directly predict the confidence/uncertainty, we compute the degree of uncertainty by calculating the entropy of the predicted probability. Although variance is wildly used in generating uncertainty for the prediction ensemble based regression models, for our binary segmentation task, we use entropy of mean prediction as confidence measure, which is more suitable. Specifically, entropy is a measure of disorder, or unpredictability. For in-distribution (ID) samples, we aim to produce concentrated predictions on one specific category, leading to low entropy. On the contrary, for out-of-distribution (OOD) samples, the model should produce uniform distribution across the categories, leading to high entropy. Accordingly, entropy score can be used to identify in-distribution samples and out-of-distribution samples.

Table 3. Performance of single model uncertainty modeling methods.

Method	DUTS [60]		ECSSD [70]		DUT [71]	
	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑
Base	0.151	0.952	0.127	0.959	0.211	0.925
Post-hoc Techniques						
MCP [22]	0.151	0.952	0.127	0.959	0.211	0.925
Energy [41]	0.151	0.950	0.151	0.949	0.192	0.932
SML [28]	0.192	0.937	0.164	0.947	0.257	0.906
GradNorm [25]	0.177	0.936	0.174	0.936	0.216	0.914
ExGrad [26]	0.151	0.952	0.127	0.959	0.211	0.925
Training Regularization Techniques						
TCP [9]	**0.126**	**0.955**	**0.109**	**0.961**	**0.166**	**0.933**
DC [58]	0.162	0.946	0.136	0.955	0.212	0.923
ReAct [54]	0.148	0.953	0.126	0.958	0.201	0.927

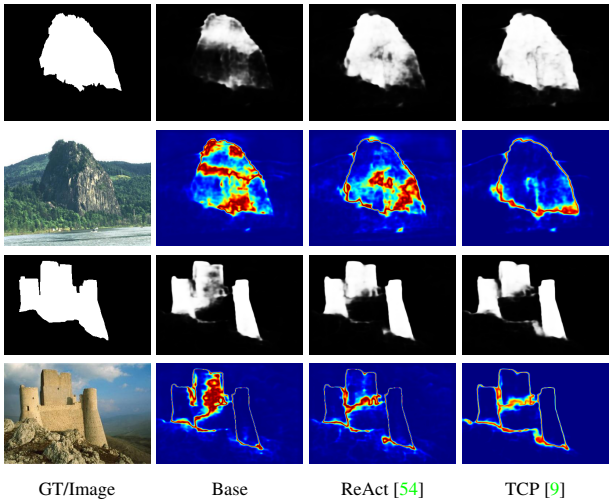

Figure 4. Qualitative comparison of single-model uncertainty modeling strategies, *i.e*. ReAct [54] and TCP [9], which both indicate more correct predictions and reliable uncertainty maps.

3.3. Performance of Distribution Gap Modeling Techniques for Saliency Detection

Conventional Solutions Analysis: Two widely studied conventional epistemic uncertainty modeling strategies are Monte Carlo (MC) dropout [15] and Deep ensemble [34]. The former is achieved via approximating a Bayesian neural network by applying dropout after the convolutional layer, which is further relaxed in [30], where they prove that adding dropout in the decoder is sufficient to achieve Bayesian approximation. Deep ensemble achieves multiple mapping functions from the input space to the output space, which is proven the most effective model uncertainty modeling techniques, although it's more computational expensive compared with the free-lunch MC-dropout technique.

The quantitative evaluation results of traditional uncertainty modeling methods (dropout and deep ensemble) are shown in Tab. 1. Compared with base model and Dropout strategy, the deep ensemble method integrates the information of multiple decoders, making it more accurate in distributional uncertainty modeling (see Fig. 3).

Long-tail Learning based Methods: We show performance of long-tail methods for SOD in Tab. 2, and observe consistent inferior performance of each solution compared with the base model. The main reason is that long-tail learning models usually rely on the size of each category as class-balance weight, where the basic assumption is that there exists no transition across categories. However, as saliency is a continuous attribute, foreground and background can be transferred flexibly. We believe the continuous adaptation of discrete long-tail methods is less effective for our task.

Single-model Uncertainty Techniques: We use multiple post-hoc and training regularization methods to implement the single model uncertainty modeling, and the results are shown in Tab. 3, where prediction accuracy is not shown as its hardly affected. We find the post-hoc techniques fail to improve the uncertainty quality. On the contrary, the training techniques, especially TCP [9] and ReAct [54] are effective in generating reliable uncertainty. We then visualize the generated uncertainty maps from TCP [9] and ReAct [54] in Fig. 4, which clearly shows the advantages for both accurate prediction and reliable uncertainty generation. Although DC [58] is efficient to generate single-model uncertainty. As it predicts uncertainty by adding random image perturbations, for pixel-level prediction, the perturbation will greatly affects the accuracy.

Test-time solutions Test-time strategies improve performance through self-supervised adaptive learning on the test set (CoTTA [61]) to make the network more adaptable to the test data, or through aggregation of multiple augmentations of the test data (CTTA [8]). The experimental results are shown in Tab. 4. We observe that the test-time training method may lead to the model that focuses on mis-classified pixels in self-supervised learning, resulting in performance degradation. In this case, carefully designed strategy to prevent model from drifting too much should be imposed. On the other hand, the test-time testing method has the potential to generate reliable uncertainty if proper augmentation policies are learned.

We also show some qualitative results in Fig. 5, and it can be seen that the use of data augmentation methods can help us explore low-density and calibrate mis-classified regions to learn the correct distributional uncertainty. However, for dense prediction tasks, due to the requirement of pixel-by-pixel correspondence, augmentations such as interception on test image that will lose pixel information and cannot be performed. Simultaneously, pixel values cannot be greatly changed, as it may lead to significant model difference. Therefore, the augmentation degree is minor to image classification task, which limits its flexibility for SOD.

Table 4. Performance of test-time strategies. CoTTA [61] is a test-time training method and CTTA [8] is a test-time augmentation technique.

Method	DUTS [60]					ECSSD [70]					DUT [71]				
	maxF_β ↑	IoU ↑	Accuracy ↑	FPR95 ↓	AUROC ↑	maxF_β ↑	IoU ↑	Accuracy ↑	FPR95 ↓	AUROC ↑	maxF_β ↑	IoU ↑	Accuracy ↑	FPR95 ↓	AUROC ↑
Base	**0.849**	**0.774**	**0.965**	0.151	0.952	**0.924**	**0.869**	**0.968**	0.127	0.959	**0.753**	**0.682**	**0.946**	0.211	0.925
CoTTA [61]	0.760	0.681	0.943	0.239	0.922	0.883	0.810	0.947	0.211	0.930	0.698	0.626	0.928	0.275	0.906
CTTA [8]	0.836	0.748	0.959	**0.141**	**0.953**	0.919	0.853	0.963	**0.117**	**0.960**	**0.753**	0.674	**0.946**	**0.193**	**0.929**

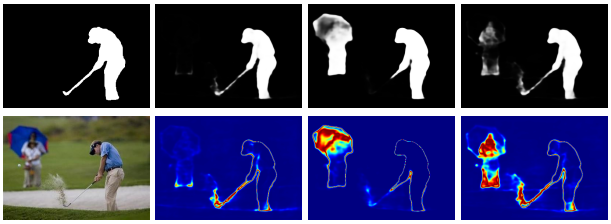

Figure 5. Visual comparison of test-time strategies, *i.e*. test-time training (CoTTA [61]) and test-time augmentation (CTTA [8]). It shows that test-time augmentation strategy can help explore low-density regions and calibrate incorrect predictions.

3.4. Analysis

MC dropout [15] is achieved by randomly dropping connections between neurons during both training and testing. Deep ensemble is designed to achieve multiple mapping functions. Our experiments show that without proper control of the dropout mask, although it's free lunch, MC dropout [15] fails to generate reliable uncertainty. Although the long-tail solutions [2, 48, 77] work nicely for class-independent classification task, we find they are less effective in modeling distributional uncertainty for our class-dependent continuous segmentation task. Single-model uncertainty is promising as it directly target distribution gap. However, the post-hoc methods rely too much [22,41] or are based on biased assumption, *i.e*. gradient based confidence estimation methods [25,26] assume uniform distribution for out-of-distribution samples, which is usually biased in practice. Training regularization methods achieve uncertainty modeling with either loss regularization [9], feature activation [54] or data regularization [58], which proved to be quite suitable for our task, as there are no class-independent assumption. The test-time strategies [8, 61] are straightforward and promising, especially test-time training. Although our current experiments fail to generate reliable uncertainty map with the current state-of-the-art test-time training techniques, we believe using suitable regularization to control the drift degree of test-time training can be promising in generating reliable distributional uncertainty. We further show the distribution of AUROC metric on DUTS testing dataset [60] in Fig. 6, which clearly shows that deep ensemble [34] and TCP [9] are effective in generating reliable distributional uncertainty.

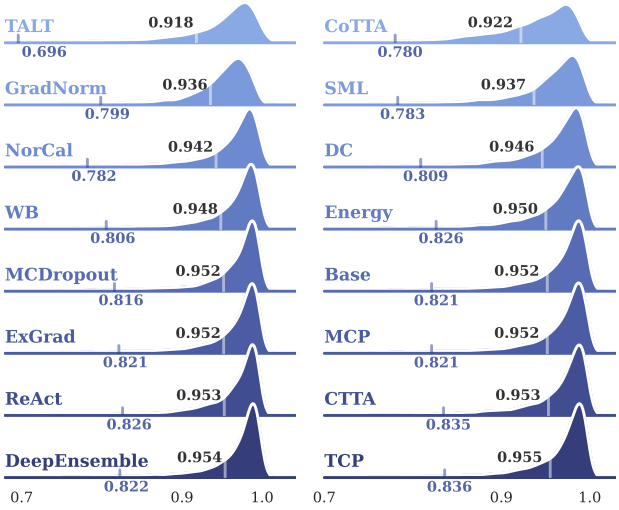

Figure 6. The distribution of AUROC metric on DUTS testing dateset [60], where x-axis is the AUROC measure, y-axis is the number of samples. The white line with the black number on the left indicates the mean AUROC of each method. The short blue line on the left represents the value of the lower 5% percentile.

4. Conclusion

Although much progress has been made to improve model performance of SOD on benchmark testing datasets, no attention has been paid to the out-of-distribution problem in SOD. In this paper, we make the first effort in investigating the out-of-distribution discovery issue for SOD, to explain the distribution gap. We perform extensive experimental results and verify the effectiveness of deep ensemble [34] and the single-model uncertainty estimation technique, *i.e*. TCP [9], in generating reliable distributional uncertainty. We find that although long-tail learning solutions are effective in class-independent classification tasks, they fail to generalize well to our class-dependent task. We also point out that although the current implementation of test-time training fails to improve uncertainty quality, it's promising to further explore regularization terms to control the drift degree of model for better uncertainty generation.

Acknowledgements

This research was supported in part by National Natural Science Foundation of China (62271410). We would like to thank the anonymous reviewers and the ACs for their useful feedback.

References

[1] Chirag Agarwal, Daniel D'souza, and Sara Hooker. Estimating example difficulty using variance of gradients. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10368–10378, 2022. 4, 6

[2] Shaden Alshammari, Yu-Xiong Wang, Deva Ramanan, and Shu Kong. Long-tailed recognition via weight balancing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6897–6907, 2022. 3, 6, 8

[3] Sherwin Bahmani, Oliver Hahn, Eduard Zamfir, Nikita Araslanov, Daniel Cremers, and Stefan Roth. Semantic self-adaptation: Enhancing generalization with a single sample. *arXiv preprint arXiv:2208.05788*, 2022. 2, 5

[4] Alexander Bartler, Andre Bühler, Felix Wiewel, Mario Döbler, and Bin Yang. Mt3: Meta test-time training for self-supervised test-time adaption. In *International Conference on Artificial Intelligence and Statistics*, pages 3080–3090, 2022. 2, 5

[5] Senqi Cao and Zhongfei Zhang. Deep hybrid models for out-of-distribution detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4733–4743, 2022. 1

[6] Hao Chen and Youfu Li. Progressively complementarity-aware fusion network for RGB-D salient object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3051–3060, 2018. 1

[7] Zhixiang Chi, Yang Wang, Yuanhao Yu, and Jin Tang. Test-time fast adaptation for dynamic scene deblurring via meta-auxiliary learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9137–9146, 2021. 2, 5

[8] Sewhan Chun, Jae Young Lee, and Junmo Kim. Cyclic test time augmentation with entropy weight method. In *Uncertainty in Artificial Intelligence*, pages 433–442, 2022. 2, 5, 7, 8

[9] Charles Corbière, Nicolas THOME, Avner Bar-Hen, Matthieu Cord, and Patrick Pérez. Addressing failure prediction by learning model confidence. In *Conference on Neural Information Processing Systems (NeurIPS)*, pages 2902–2913, 2019. 2, 4, 7, 8

[10] Jiequan Cui, Yuhui Yuan, Zhisheng Zhong, Zhuotao Tian, Han Hu, Stephen Lin, and Jiaya Jia. Region rebalance for long-tailed semantic segmentation. *arXiv preprint arXiv:2204.01969*, 2022. 2

[11] Deng-Ping Fan, Yingjie Zhai, Ali Borji, Jufeng Yang, and Ling Shao. BBS-Net: RGB-D salient object detection with a bifurcated backbone strategy network. In *European Conference on Computer Vision (ECCV)*, pages 275–292, 2020. 1

[12] Ke Fan, Yikai Wang, Qian Yu, Da Li, and Yanwei Fu. A simple test-time method for out-of-distribution detection. *arXiv preprint arXiv:2207.08210*, 2022. 2, 5

[13] Chengjian Feng, Yujie Zhong, and Weilin Huang. Exploring classification equilibrium in long-tailed object detection. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3417–3426, 2021. 2

[14] Keren Fu, Deng-Ping Fan, Ge-Peng Ji, and Qijun Zhao. JL-DCF: Joint learning and densely-cooperative fusion framework for RGB-D salient object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3052–3062, 2020. 1

[15] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning (ICML)*, pages 1050–1059, 2016. 2, 6, 7, 8

[16] Yossi Gandelsman, Yu Sun, Xinlei Chen, and Alexei A. Efros. Test-time training with masked autoencoders. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022. 2, 5

[17] Will Grathwohl, Kuan-Chieh Wang, Joern-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations (ICLR)*, 2020. 4

[18] Matej Grcić, Petra Bevandić, and Siniša Šegvić. Dense-hybrid: Hybrid anomaly detection for dense open-set recognition. In *European Conference on Computer Vision (ECCV)*, pages 500–517, 2022. 1

[19] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning (ICML)*, pages 1321–1330, 2017. 2

[20] Nicklas Hansen, Rishabh Jangir, Yu Sun, Guillem Alenyà, Pieter Abbeel, Alexei A Efros, Lerrel Pinto, and Xiaolong Wang. Self-supervised policy adaptation during deployment. *International Conference on Learning Representations (ICLR)*, 2021. 2, 5

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 2

[22] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations (ICLR)*, 2017. 2, 4, 7, 8

[23] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012. 3

[24] Julia Hornauer and Vasileios Belagiannis. Gradient-based uncertainty for monocular depth estimation. In *European Conference on Computer Vision (ECCV)*, pages 613–630, 2022. 4

[25] Rui Huang, Andrew Geng, and Yixuan Li. On the importance of gradients for detecting distributional shifts in the wild. In *Conference on Neural Information Processing Systems (NeurIPS)*, pages 677–689, 2021. 2, 4, 7, 8

[26] Conor Igoe, Youngseog Chung, Ian Char, and Jeff Schneider. How useful are gradients for ood detection really? *arXiv preprint arXiv:2205.10439*, 2022. 4, 7, 8

[27] Sergey Ioffe. Batch renormalization: Towards reducing minibatch dependence in batch-normalized models. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017. 2

[28] Sanghun Jung, Jungsoo Lee, Daehoon Gwak, Sungha Choi, and Jaegul Choo. Standardized max logits: A simple yet effective approach for identifying unexpected road obstacles in urban-scene segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, pages 15425–15434, 2021. 4, 7

[29] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *International Conference on Learning Representations (ICLR)*, 2019. 2

[30] 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. 7

[31] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Conference on Neural Information Processing Systems (NeurIPS)*, volume 30, 2017. 2

[32] Ildoo Kim, Younghoon Kim, and Sungwoong Kim. Learning loss for test-time augmentation. In *Conference on Neural Information Processing Systems (NeurIPS)*, volume 33, pages 4163–4174, 2020. 2, 5

[33] Masanari Kimura. Understanding test-time augmentation. In *International Conference on Neural Information Processing*, pages 558–569, 2021. 2, 5

[34] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Conference on Neural Information Processing Systems (NeurIPS)*, volume 30, 2017. 2, 6, 7, 8

[35] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and Fujie Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006. 4

[36] Jinsol Lee and Ghassan AlRegib. Gradients as a measure of uncertainty in neural networks. In *IEEE International Conference on Image Processing (ICIP)*, pages 2416–2420, 2020. 4

[37] Bo Li, Yongqiang Yao, Jingru Tan, Gang Zhang, Fengwei Yu, Jianwei Lu, and Ye Luo. Equalized focal loss for dense long-tailed object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6990–6999, 2022. 2

[38] Jiale Li, Hang Dai, and Yong Ding. Self-distillation for robust LiDAR semantic segmentation in autonomous driving. In *European Conference on Computer Vision (ECCV)*, pages 659–676, 2022. 2, 5

[39] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations (ICLR)*, 2018. 4

[40] Nian Liu, Ni Zhang, Kaiyuan Wan, Ling Shao, and Junwei Han. Visual saliency transformer. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4722–4732, 2021. 1

[41] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. In *Conference on Neural Information Processing Systems (NeurIPS)*, volume 33, pages 21464–21475, 2020. 4, 7, 8

[42] Helen Lu, Divya Shanmugam, Harini Suresh, and John Guttag. Improved text classification via test-time augmentation. *arXiv preprint arXiv:2206.13607*, 2022. 2, 5

[43] Alexander Lyzhov, Yuliya Molchanova, Arsenii Ashukha, Dmitry Molchanov, and Dmitry Vetrov. Greedy policy search: A simple baseline for learnable test-time augmentation. In *Conference on Uncertainty in Artificial Intelligence*, pages 1308–1317, 2020. 2, 5

[44] Yuxin Mao, Jing Zhang, Zhexiong Wan, Yuchao Dai, Aixuan Li, Yunqiu Lv, Xinyu Tian, Deng-Ping Fan, and Nick Barnes. Generative transformer for accurate and reliable salient object detection. *arXiv preprint arXiv:2104.10127*, 2021. 1

[45] Vaishnavh Nagarajan, Anders Andreassen, and Behnam Neyshabur. Understanding the failure modes of out-of-distribution generalization. In *International Conference on Learning Representations (ICLR)*, 2021. 2

[46] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Hybrid models with deep and invertible features. In *International Conference on Machine Learning (ICML)*, pages 4723–4732, 2019. 1

[47] Kemal Oksuz, Baris Can Cam, Sinan Kalkan, and Emre Akbas. Imbalance problems in object detection: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 43(10):3388–3415, 2020. 2

[48] Tai-Yu Pan, Cheng Zhang, Yandong Li, Hexiang Hu, Dong Xuan, Soravit Changpinyo, Boqing Gong, and Wei-Lun Chao. On model calibration for long-tailed object detection and instance segmentation. In *Conference on Neural Information Processing Systems (NeurIPS)*, pages 2529–2542, 2021. 3, 6, 8

[49] Juan C. Pérez, Motasem Alfarra, Guillaume Jeanneret, Laura Rueda, Ali Thabet, Bernard Ghanem, and Pablo Arbeláez. Enhancing adversarial robustness via test-time transformation ensembling. In *IEEE International Conference on Computer Vision (ICCV) Workshop*, pages 81–91, 2021. 2, 5

[50] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 44(3):1623–1637, 2020. 2

[51] Aneeshan Sain, Ayan Kumar Bhunia, Vaishnav Potlapalli, Pinaki Nath Chowdhury, Tao Xiang, and Yi-Zhe Song. Sketch3t: Test-time training for zero-shot sbir. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7462–7471, 2022. 2, 5

[52] Anindya Sarkar, Anirban Sarkar, and Vineeth N Balasubramanian. Leveraging test-time consensus prediction for robustness against unseen noise. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1839–1848, 2022. 2, 5

[53] Divya Shanmugam, Davis W. Blalock, Guha Balakrishnan, and John V. Guttag. Better aggregation in test-time augmentation. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1194–1203, 2021. 2, 5

[54] Yiyou Sun, Chuan Guo, and Yixuan Li. React: Out-of-distribution detection with rectified activations. In *Conference on Neural Information Processing Systems (NeurIPS)*, volume 34, pages 144–157, 2021. 2, 4, 7, 8

[55] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning (ICML)*, pages 9229–9248, 2020. 2, 5

[56] Jingru Tan, Xin Lu, Gang Zhang, Changqing Yin, and Quanquan Li. Equalization loss v2: A new gradient balance approach for long-tailed object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1685–1694, 2021. 2

[57] Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Equalization loss for long-tailed object recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11662–11671, 2020. 2

[58] Jayaraman J Thiagarajan, Rushil Anirudh, Vivek Narayanaswamy, and Peer-Timo Bremer. Single model uncertainty estimation via stochastic data centering. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022. 4, 7, 8

[59] Jiaqi Wang, Wenwei Zhang, Yuhang Zang, Yuhang Cao, Jiangmiao Pang, Tao Gong, Kai Chen, Ziwei Liu, Chen Change Loy, and Dahua Lin. Seesaw loss for long-tailed instance segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9695–9704, 2021. 2

[60] Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. Learning to detect salient objects with image-level supervision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 136–145, 2017. 5, 6, 7, 8

[61] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7201–7211, 2022. 2, 5, 7, 8

[62] Tao Wang, Yu Li, Bingyi Kang, Junnan Li, Junhao Liew, Sheng Tang, Steven Hoi, and Jiashi Feng. The devil is in classification: A simple framework for long-tail instance segmentation. In *European Conference on Computer Vision (ECCV)*, pages 728–744, 2020. 2

[63] Tong Wang, Yousong Zhu, Chaoyang Zhao, Wei Zeng, Jinqiao Wang, and Ming Tang. Adaptive class suppression loss for long-tail object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3103–3112, 2021. 2

[64] Jun Wei, Shuhui Wang, and Qingming Huang. F^3net: Fusion, feedback and focus for salient object detection. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 12321–12328, 2020. 1

[65] Jun Wei, Shuhui Wang, Zhe Wu, Chi Su, Qingming Huang, and Qi Tian. Label decoupling framework for salient object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13025–13034, 2020. 1

[66] Qitian Wu, Hengrui Zhang, Junchi Yan, and David Wipf. Handling distribution shifts on graphs: An invariance perspective. In *International Conference on Learning Representations (ICLR)*, 2022. 2

[67] Yu-Huan Wu, Yun Liu, Le Zhang, Ming-Ming Cheng, and Bo Ren. Edn: Salient object detection via extremely-downsampled network. *IEEE Transactions on Image Processing (TIP)*, 31:3125–3136, 2022. 1

[68] Zhenyu Wu, Shuai Li, Chenglizhao Chen, Aimin Hao, and Hong Qin. Recursive multi-model complementary deep fusion for robust salient object detection via parallel sub-networks. *Pattern Recognition (PR)*, 121:108212, 2022. 1

[69] Binwei Xu, Haoran Liang, Ronghua Liang, and Peng Chen. Locate globally, segment locally: A progressive architecture with knowledge review network for salient object detection. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 3004–3012, 2021. 1

[70] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1155–1162, 2013. 5, 6, 7, 8

[71] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3166–3173, 2013. 5, 6, 7, 8

[72] Maoke Yang, Kun Yu, Chi Zhang, Zhiwei Li, and Kuiyuan Yang. Denseaspp for semantic segmentation in street scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3684–3692, 2018. 2

[73] Hongjie Zhang, Ang Li, Jie Guo, and Yanwen Guo. Hybrid models for open set recognition. In *European Conference on Computer Vision (ECCV)*, pages 102–117, 2020. 1

[74] Jing Zhang, Deng-Ping Fan, Yuchao Dai, Saeed Anwar, Fatemeh Sadat Saleh, Sadegh Aliakbarian, and Nick Barnes. Uncertainty inspired rgb-d saliency detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 44(9):5761–5779, 2021. 1

[75] Jing Zhang, Nick Barnes Jianwen Xie, and Ping Li. Learning generative vision transformer with energy-based latent space for saliency prediction. In *Conference on Neural Information Processing Systems (NeurIPS)*, volume 34, pages 15448–15463, 2021. 1

[76] Jing Zhang, Xin Yu, Aixuan Li, Peipei Song, Bowen Liu, and Yuchao Dai. Weakly-supervised salient object detection via scribble annotations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12546–12555, 2020. 1

[77] Yifan Zhang, Bryan Hooi, Lanqing Hong, and Jiashi Feng. Self-supervised aggregation of diverse experts for test-agnostic long-tailed recognition. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022. 2, 3, 6, 8

[78] Xiaoqi Zhao, Youwei Pang, Lihe Zhang, Huchuan Lu, and Lei Zhang. Suppress and balance: A simple gated network for salient object detection. In *European Conference on Computer Vision (ECCV)*, pages 35–51, 2020. 1