Uncertainty-aware Salient Object Detection

Jing Zhang

A thesis submitted for the degree of
Doctor of Philosophy at
The Australian National University

October 2021

Except where otherwise indicated, this thesis is my own original work.

Jing Zhang
3 October 2021

Thesis Outcome

Publications

1. **Jing Zhang**, Jianwen Xie, Nick Barnes, Ping Li. Learning Generative Vision Transformer with Energy-Based Latent Space for Saliency Prediction. NeurIPS 2021.

2. **Jing Zhang**, Deng-Ping Fan, Yuchao Dai, Xin Yu, Yiran Zhong, Nick Barnes, Ling Shao. RGB-D Saliency Detection via Cascaded Mutual Information Minimization. ICCV 2021.

3. **Jing Zhang**, Deng-Ping Fan, Yuchao Dai, Saeed Anwar, Fatemeh Sadat Saleh, Sadegh Aliakbarian, Nick Barnes. Uncertainty Inspired RGB-D Saliency Detection. TPAMI 2021.

4. Yunqiu Lv*, **Jing Zhang***, Yuchao Dai, Aixuan Li, Bowen Liu, Nick Barnes, Deng-Ping Fan. Simultaneously Localize, Segment and Rank the Camouflaged Objects. CVPR 2021 (*: Equal Contribution).

5. Aixuan Li*, **Jing Zhang***, Yunqiu Lv, Bowen Liu, Tong Zhang, Yuchao Dai. Simultaneously Localize, Uncertainty-aware Joint Salient Object and Camouflaged Object Detection. CVPR 2021 (*: Equal Contribution).

6. **Jing Zhang**, Yuchao Dai, Tong Zhang, Mehrtash Harandi, Nick Barnes, Richard Hartley. Learning Saliency from Single Noisy Labelling: A Robust Model Fitting Perspective. TPAMI 2020.

7. **Jing Zhang**, Deng-Ping Fan, Yuchao Dai , Saeed Anwar, Fatemeh Sadat Saleh, Tong Zhang, Nick Barnes. UC-Net: Uncertainty Inspired RGB-D Saliency Detection via Conditional Variational Autoencoders. CVPR 2020.

8. **Jing Zhang**, Xin Yu, Aixuan Li, Peipei Song, Bowen Liu, Yuchao Dai. Weakly-Supervised Salient Object Detection via Scribble Annotations. CVPR 2020.

9. **Jing Zhang**, Nick Barnes, Jianwen Xie. Learning Noise-Aware Encoder-Decoder from Noisy Labels by Alternating Back-Propagation for Saliency Detection. ECCV 2020.

10. **Jing Zhang**, Tong Zhang, Yuchao Dai, Mehrtash Harandi, Richard Hartley. Deep unsupervised saliency detection: A multiple noisy labeling perspective. CVPR 2018.

To my dearest family for their love and support.

Acknowledgments

Significant courage is needed for one to start the PhD journey. I'm fortunate to have love and support from both my family and my friends, which helps me grow much stronger and become a better person.

I would first thank my primary supervisor, Professor Nick Barnes. Nick is kind and always show passion for research. He respects my ideas, and helps me polish them to be published. I deeply appreciate all the help and understanding from him. I also want to thank my chair supervisor, Professor Richard Hartley. Richard is quite strict about mathematics, which inspires me to think deeper about my research instead of following the conventional solutions as others. Meanwhile, I want to thank my other supervisors, Professor Yuchao Dai and Dr. Mehrtash Harandi. Yuchao introduced the saliency detection task to me, which is my main direction for the Ph.D degree. He helped me in most of my publications by formalizing the ideas and writing the papers. Mehrtash helps me to think in both computer vision way and machine learning manner, which inspires most of my recent work of combining machine learning idea in the computer vision tasks.

I had a three months internship with Dr. Jianwen Xie at Hikvision USA in 2019. I'm grateful that Jianwen chose me instead of other competitive candidates. I firmly believe that he greatly shape my current understanding about my research. Until now, we still closely work together, and I quite enjoy learning from him.

I met Dr. Dengping Fan in CVPR 2019, and we then co-work for several papers. He is a very kind person, and also a very active researcher. He teaches me how to write a good paper, and shares with me his experience of doing research. I appreciate it a lot that he is trying his best to make me a stronger researcher.

I am fortunate to have worked with my colleagues, including Xin Yu, Fatemeh Sadat Saleh, Saeed Anwar, Sadegh Aliakbarian, Liu Liu, Liyuan Pan, Jun Zhang, Yiran Zhong, Kaihao Zhang, Hongguang Zhang, Tong Zhang, Haodong Yao, Yue Cao, Yujiao Shi, Yao Lu, Wei Mao, Dongxu Li, Shihao Jiang, Ziang Chen, Kaiyue Lu, Peipei Song, Weixuan Sun, Changkun Ye. I also want to thank my friends: Xiaofeng Li, Qingtan Wang, Ruirui Yao, Sally Wang, Jing Wei, Wenqiang Duan.

Last but not the least, I would also like to express my deep acknowledgement to my family and I dedicate this thesis to my whole family.

Abstract

Saliency detection models are trained to discover the region(s) of an image that attract human attention. According to whether depth data is used, static image saliency detection models can be divided into RGB image saliency detection models, and RGB-D image saliency detection models. The former predict salient regions of the RGB image, while the latter take both the RGB image and the depth data as input. Conventional saliency prediction models (both RGB saliency and RGB-D saliency) typically learn a deterministic mapping from images to the corresponding ground truth saliency maps without modeling the uncertainty of predictions, following the supervised learning pipeline. This thesis is dedicated to learning a conditional distribution over saliency maps, given an input image (or image pair for RGB-D data), and modeling the uncertainty of predictions.

For RGB-D saliency detection, we present the first generative model based framework to achieve uncertainty-aware prediction. Our framework includes two main models: 1) a generator model, which maps the input image and latent variable to stochastic saliency prediction, and 2) an inference model, which gradually updates the latent variable by sampling it from the true or approximate posterior distribution. The generator model is an encoder-decoder saliency network. To infer the latent variable, we introduce two different solutions: i) a Conditional Variational Auto-encoder with an extra encoder to approximate the posterior distribution of the latent variable; and ii) an Alternating Back-Propagation technique, which directly samples the latent variable from the true posterior distribution.

One drawback of the above models is that it fails to explicitly model the connection between RGB image and depth data to achieve effective cooperative learning. We further introduce a novel latent variable model based complementary learning framework to explicitly model the complementary information between the two modes, namely the RGB mode and depth mode. Specifically, we first design a regularizer using mutual-information minimization to reduce the redundancy between appearance features from RGB and geometric features from depth in the latent space. Then we fuse the latent features of each mode to achieve multi-modal feature fusion. Extensive experiments on benchmark RGB-D saliency datasets illustrate the effectiveness of our framework.

For RGB saliency detection, we propose a generative saliency prediction model based on the conditional generative cooperative network, where a conditional latent variable model and a conditional energy-based model are jointly trained to predict saliency in a cooperative manner. The latent variable model serves as a coarse saliency model to produce a fast initial prediction, which is then refined by Langevin revision of the energy-based model that serves as a fine saliency model. We call this probabilistic coarse-to-fine saliency prediction.

Apart from the fully supervised learning framework, we also investigate weakly supervised learning, and propose the first scribble-based weakly-supervised salient object detection model. In doing so, we first relabel an existing large-scale salient object detection dataset with scribbles, namely S-DUTS dataset. Since object structure and detail information is not identified by scribbles, directly training with scribble labels will lead to saliency maps of poor boundary localization. To mitigate this problem, we propose an auxiliary edge detection task to localize object edges explicitly, and a gated structure-aware loss to place constraints on the scope of structure to be recovered.

To further reduce the labeling burden, we introduce a noise-aware encoder-decoder framework to disentangle a clean saliency predictor from noisy training examples, where the noisy labels are generated by unsupervised handcrafted feature-based methods. The proposed model consists of two sub-models parameterized by neural networks: (1) a saliency predictor that maps input images to clean saliency maps, and (2) a noise generator, which is a latent variable model that produces noise from Gaussian latent vectors. The whole model that represents noisy labels is a sum of the two sub-models. The goal of training the model is to estimate the parameters of both sub-models, and simultaneously infer the corresponding latent vector of each noisy label. We propose to train the model by using an alternating back-propagation algorithm, which alternates the following two steps: (1) learning back-propagation for estimating the parameters of two sub-models by gradient ascent, and (2) inferential back-propagation for inferring the latent vectors of training noisy examples by Langevin Dynamics. To prevent the network from converging to trivial solutions, we utilize an edge-aware smoothness loss to regularize hidden saliency maps to have similar structures as their corresponding images.

Following a similar setting, we propose to learn saliency from a single noisy labelling, and exploit model consistency across iterations to identify inliers and outliers (i.e., , noisy labels). Extensive experiments on different benchmark datasets demonstrate the superiority of our proposed framework, which can learn comparable saliency prediction with state-of-the-art fully supervised saliency methods. Furthermore, we show that simply by treating ground truth annotations as noisy labelling, our framework achieves tangible improvements over state-of-the-art methods.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

When viewing a scene, the human visual system has the ability to selectively locate attention [2, 3, 4, 5, 6] on informative contents, which locally stand out from their surroundings. This selection is usually performed in the form of a spatial circumscribed region, leading to the so-called "focus of attention" [5], which scans the scene rapidly in a bottom-up and task-independent manner or slowly in a top-down task-dependent manner. [2] introduced a general attention model to explain the human visual search strategies [7]. Specifically, the visual input is first decomposed into a group of topographic feature maps which they defined as the early representations. Then, different spatial locations compete for saliency within each topographic feature map, such that locations that locally stand out from their surrounding persist. Lastly, all the feature maps are fed into a master "saliency map", indicating the topographically codes for saliency over the visual scene [5].

Following the above process of saliency selection, early saliency detection models focus on detecting the informative locations, leading to the eye fixation prediction [8] task, which aims to find the informative locations without preserving the semantic structure information. [9] and [10] then extended the salient locations driven methods [2, 5] and introduced the salient object detection task, which is a binary segmentation task aiming to identify the full scope of the salient object. In this way, "salient object" is defined as any item that is distinct from those around it. Many factors can lead something to be "salient", including the stimulus itself that makes the item distinct, i.e., color, texture, direction of movement and *etc.* , and the internal cognitive state of the observer, leading to his/her understanding of saliency. In this thesis, we discuss only the salient object detection to produce saliency map highlighting the scope of the salient object.

As an important computer vision task, salient object detection is intrinsic to various tasks such as image cropping [11], context-aware image editing [12], image recognition [13], interactive image segmentation [14], action recognition [15], image caption generation [16] and semantic image labeling [17], where saliency models can be used to extract class-agnostic important areas in an image or a video sequence. Although considerable progress has been achieved, it still remains as a challenging task and requires effective approaches to handle complex real-world scenarios.

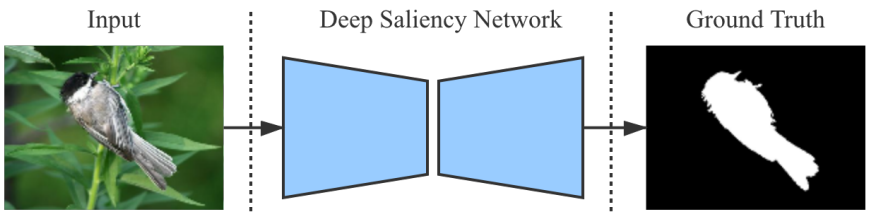


Figure 1.1: The general pipeline of a fully supervised saliency detection network.

Conventional saliency detection methods either employ predefined features such as color and texture descriptors [18][19], or indicators of appearance uniqueness [20] and region compactness [21] based on specific statistical priors such as center prior [22], contrast prior [23], boundary prior [24] and object prior [25]. One typical direction is computing the pixel-wise or superpixel-wise saliency directly from the above handcrafted features [24, 26]. There also exist supervised models [23, 18] that are trained with the pre-defined features. Due to the limited representation ability of the handcrafted features, these conventional methods achieve acceptable results only on relatively simple datasets (see [27] for a dedicated survey on saliency detection prior to the deep learning revolution), but their performances deteriorate quickly when the input images become cluttered and complicated.

Deep learning with convolutional neural networks (CNNs) has obtained great success in many vision tasks, such as image classification [28] and semantic segmentation [29]. The success has also been extended to the task of saliency detection [30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40], where the problem is generally formulated as a dense labeling task that automatically learns feature representations of salient regions, outperforming handcrafted feature based solutions with a wide margin.

The de-facto standard for those deep learning based techniques is to train a deep neural network using ground truth (GT) saliency maps provided by the corresponding benchmark datasets as shown in Fig. 1.1, where the GT saliency maps are obtained through human consensus or by the dataset creators [41, 42, 43]. Building upon large scale saliency datasets, deep convolutional neural network based saliency prediction models [33, 34, 37, 44, 45, 46, 47, 1, 48, 49] have made profound progress in learning the mapping from the input image to the corresponding GT saliency map.

As discussed above, to model human visual attention in the deep learning era, ground truth saliency maps are provided as an approximation of visual perception saliency. In this way, multiple labelers are instructed to find the key objects corresponds to saliency. To provide a binary saliency map, "majority voting" is usually performed, which then defines the majority salient regions as being salient foreground in the final ground truth saliency map. In Fig. 1.2, we show the provided salient foreground region after majority voting (the object(s) with red rectangle(s)) and other candidate salient regions that are discarded after majority voting (the ob-

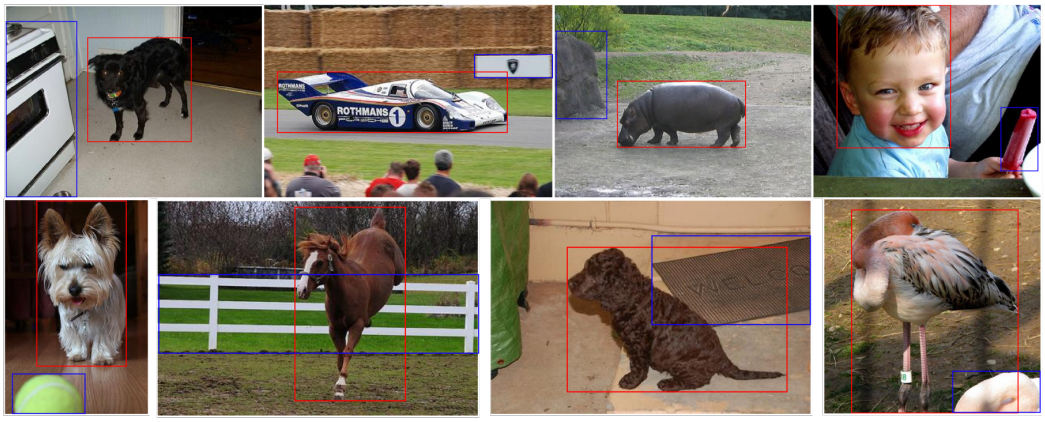


Figure 1.2: The subjective nature of saliency.

ject(s) with blue rectangle(s)). It clearly shows that although the majority regions represent the most salient regions, there still exist other candidate salient regions that may attract human attention. We then argue that the existing "majority voting ground truth" based saliency models focus only on the "most" salient regions, which can be biased and fail to discover the whole story of human visual perception saliency.

Instead of obtaining a deterministic saliency prediction model that estimates the most salient regions of the input image, in this thesis, we would like to produce a series of predictions covering the possible salient regions following human visual perception saliency. To do so, we first formulate the saliency prediction as a two-step task: 1) coarse saliency prediction to identify the salient objects in a gross sense; 2) fine-scale salient segmentation to find the whole scope of the salient objects. Then, we propose to achieve distribution estimation that can produce all possible salient regions for each input image with the trained statistics representing the saliency distribution, e.g., mean value μ indicating average saliency estimation and standard deviation σ measuring the uncertainty of prediction.

To estimate the distribution of saliency predictions, one can employ a generative model, which is a powerful way of learning the data distribution. As a special case of generative models, latent variable models, e.g., Variational Autoencoder (VAE) [50, 51], Generative Adversarial Network (GAN) [52], introduce a latent variable to the network representing randomness or uncertainty of prediction. In the VAE based framework, the latent variable is assumed to follow a specific Gaussian distribution with a prior distribution of a standard normal distribution. There exists an inference model to map the input image to the latent variable, representing the posterior distribution of the latent variable. The network is trained to minimize both the reconstruction loss (the difference of the reconstructed image and the raw input image) and a regularizer that penalizes the distance between the prior and posterior distribution of the latent variable. In GAN related frameworks, there exists no inference model, and one can directly sample the latent variable from the fixed prior distribution,

e.g., standard normal distribution. Another approach to estimate the distribution of predictions is the energy-based model (EBM) [53, 54, 55, 56, 57, 58, 59], which is represented as a deep neural network to learn an energy function and assigns low energy to in-distribution samples and vice versa.

Furthermore, we observe that the success of the above fully supervised models highly depends on the availability of large scale pixel-wise annotations, which are both expensive and time-consuming to obtain. For example, the most widely used saliency detection training dataset, namely DUTS [41], includes 10,553 training images, which takes more than three months to label, and costs more than $10,000. To relieve the burden of pixel-wise labeling, we also work on weakly-supervised and unsupervised salient object detection. For the former, we aim at producing easy-to-access labels instead of the pixel-wise annotation. For the latter, we further reduce the labeling effort by directly computing the noisy saliency maps with conventional handcrafted feature based solutions.

In this thesis, we introduce the first generative model based saliency detection network with both the latent variable model [60, 61] and the energy-based model [62], to produce a distribution of prediction instead of the point estimation as performed by existing techniques. With our solution, we can produce multiple predictions for each input image, and we then define the variance of those multiple predictions as uncertainty, representing confidence of prediction.

Apart from the probabilistic network, we also design weakly-supervised [63] and unsupervised [64, 65, 66] learning frameworks to learn saliency from easy-to-access weak labels, namely scribble annotation based saliency detection network and noisy label based learning strategies.

In the following, we first briefly introduce the main techniques we adopted in this thesis in Section 1.2, including generative models, weakly supervised learning and learning from noisy labeling. Then we introduce our main contributions in Section 1.3. Note that, although the proposed solutions are for saliency detection task, we claim that our solutions are general ideas, and they can be easily extend to other dense prediction tasks. We introduce the possible extension of our solutions in Chapter 8.

1.2 Related Techniques

Here we introduce related techniques to the high level themes of our work, which will be discussed in detail in the corresponding chapters.

1.2.1 Generative Model

Different from deterministic models [28, 67] which directly achieve a mapping with the provided training dataset, generative models [50, 51, 52, 56] learn an underlying data distribution by analyzing the given training dataset as shown in Fig. 1.3. Once trained, generative models can then produce new samples to match the data distribution. In this way, we define deterministic models as point estimation techniques,

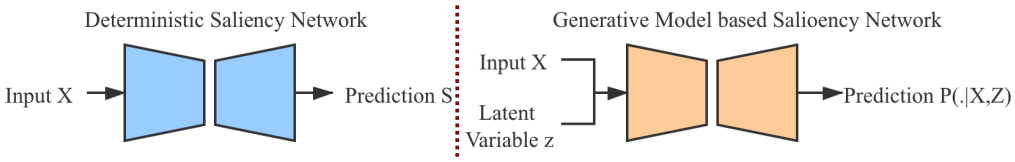


Figure 1.3: The framework difference of the deterministic saliency network and generative model based saliency network.

and generative models as distribution estimation solutions. Two different generative models are considered in this thesis: 1) a latent variable model and 2) an energy-based model, where the former introduces an extra latent variable to account for the uncertainty of prediction, and the latter models the distribution of the dataset and generates new samples with a similar distribution.

Latent Variable Model:

Latent variable models are statistical models that relate observed variables (e.g., input X and output Y) to the latent variables (e.g., z), which form a class of generative models that can infer the hidden structure of the provided data, e.g., $p(y|x,z)$. Given an input and output pair X, Y, deterministic models achieve point estimation $Y = f_\omega(X)$, where ω represents the mapping function, e.g., for image segmentation, ω is the parameter set that achieves a mapping from input image X to the segmentation map Y. Latent variable models estimate a distribution $p_\theta(Y|X)$ with a generator network, where θ is a parameter set of the generator network. Two main methods have been widely studied for training generative models, especially conditional generative models: 1) Variational Autoencoders (VAE) and Conditional Variational Autoencoders (CVAE) [50, 51]; and 2) Generative Adversarial Networks (GAN) and Conditional Generative Adversarial Networks (CGAN) [52, 68]. A typical VAE consists of an encoder, a decoder, and a loss function. The encoder is a neural network parameterized with weights and biases θ, which maps the input X to a latent (hidden) representation z. The decoder is another neural network with weights and biases ϕ, which reconstructs the data from z. To train a VAE, one aims to maximize the evidence lower bound of log-likelihood of the data with a reconstruction loss and a KL loss term as regularizer, where the former measures how the prediction is similar with provided ground truth, and the latter measures the ignorance about the prior distribution of the latent variable. Different from VAEs, where an extra encoder makes it possible to infer the latent variable, GANs have no inference model, and the discriminator is designed to check whether the prediction from generator is real or fake. Both models involve an extra network (inference network for VAE based model, or discriminator network for GANs) in addition to the task-related network.

In low-level vision, VAEs have been applied to tasks such as image background modeling [69], latent representations with sharp samples [70], difference of motion modes [71], medical image segmentation model [72], and modeling inherent ambiguities of an image [73]. VAEs have also been explored in more complex vision

tasks such as uncertain future forecast [74, 75], human motion prediction [76], and shape-guided image generation [77]. Recently, VAE algorithms have been extended to the 3D domain targeting applications such as 3D mesh deformation [78], and point cloud instance segmentation [79]. Similarly, the existing GAN-based generative models [80, 81, 82, 83, 84, 85, 86] usually use GANs to detect higher-order inconsistency between ground truth and the prediction.

Energy-based Model:

Energy-based generative models [57, 87, 88, 89, 59, 90, 91, 92, 58, 93, 94, 95, 96] define an unnormalized density of a high-dimensional random variable of interest, which is in the form of the exponential of the negative energy function parameterized by a neural network. [55] defined energy as a composition of latent and observable variables, while [97] designed a mapping function with EBM, where the input is directly mapped to the output space. Conventionaly, EBMs rely on stochastic gradient-descent (SGD) optimization methods that are difficult to apply to high-dimensional datasets, as the maximum likelihood learning of the energy-based model typically requires MCMC sampling, which is computational challenging. Recently, [59, 58] introduced Gradient based MCMC (Langevin Dynamics) for efficient and stable sampling, which makes it possible to scale EBMs to high-dimensional domains.

Our contributions: We explore generative models for saliency detection from two perspectives: 1) a generative model is adopted to estimate the labeling noise [65], where a generative model based noise estimation module is learned to estimate labeling noise and the latent clean label; 2) with clean labels available, the generative model is used to model the "subjective nature" of saliency [60, 61], where the latent variable is used to model the attributes of saliency that may not exist in the given training dataset. We claim that a generative model based saliency prediction network can not only provide robust prediction towards labeling noise, but also is suitable to achieve stochastic predictions, making it possible to estimate uncertainty within the trained network.

1.2.2 Weakly Supervised Learning

Instead of using the pixel-wise annotations, weakly supervised learning models [41, 98, 99, 63] rely only on easy-to-access annotations, e.g., bounding box, scribble, point supervision, image-level labels [41, 98] as shown in Fig. 1.4. As no accurate structure information is provided, weakly supervised learning models often focus on structure-recovery strategies. For example, [41] adopts a DenseCRF [29] to constrain the structure of prediction, and [63] uses auxiliary edge detection branch and structure-aware loss to produce sharp predictions. In this section, we introduce existing techniques for weakly supervised learning, including weakly supervised saliency detection and weakly supervised semantic segmentation.

Weakly Supervised Saliency Detection

Starting with the weak annotations, especially image-level annotations as shown in Fig. 1.4, existing weakly supervised saliency detection models follow a three-step learning pipeline: 1) obtaining a class activation map (CAM) [100] with the pro-

Figure 1.4: Comparison of different types of annotations.

vided image-level annotation; 2) refining the CAM with pair-wise constraints, e.g., DenseCRF [29], to produce sharp saliency prediction; 3) treating the sharp saliency prediction as pseudo label and iteratively train the model. Following the above learning pipeline, Wang *et al.* [41] introduced a foreground inference network to produce saliency maps with image-level labels. Hsu *et al.* [101] presented a category-driven map generator to learn saliency from class activation maps. Li *et al.* [98] adopted an iterative learning strategy to update an initial saliency map generated from unsupervised saliency methods by learning with image-level supervision. A fully connected CRF [29] was utilized in [41, 98] as post-processing to refine the produced saliency map. Zeng *et al.* [102] propose training saliency models with diverse weak supervision sources, including category labels, captions, and unlabeled data.

Weakly-Supervised Semantic Segmentation

Different from weakly supervised saliency detection, which mainly define image-level annotation as weak labels, the weakly supervised semantic segmentation task explores various types of weak annotations. Dai *et al.* [103] and Khoreva [104] first treat the bounding box annotation as a pseudo label, and then they iteratively updated network parameters and pseudo labels until reaching a pre-defined maximum epoch. Shi *et al.* [105] tackled the weakly-supervised semantic segmentation problem by using multiple dilated convolutional blocks of different dilation rates to encode dense object localization. Li *et al.* [106] presented an iterative bottom-up and top-down semantic segmentation framework to alternatingly expand object regions and optimize segmentation network with image tag supervision. Huang *et al.* [107] introduced a seeded region growing technique to learn semantic segmentation with image-level labels. Vernaza *et al.* [108] designed a random walk based label propagation method to learn semantic segmentation from sparse annotations.

Recovering Structure from Weak Labels

As weak annotations do not contain complete semantic regions of specific objects, the predicted object structure is often incomplete, e.g., the prediction is usually too smooth with limited structure information. To preserve rich and fine-detailed semantic information, additional regularizers [109] are often employed to set constraints on the structure of the prediction. Two main solutions are widely studied, including graph model based methods (e.g., fully connected Conditional Random Field (CRF) [29]) and boundary based losses [110]. The former recover structure information with pair-wise term, and the latter defines self-supervised loss to force the structure of the prediction to be well-aligned with image edges. Tang *et al.* [111] introduced a normal-

ized cut loss as a regularizer with partial cross-entropy loss for weakly-supervised image segmentation. Obukhov *et al.* [112] proposed a gated CRF loss to recover the structure information. Lampert *et al.* [110] presented a constrain-to-boundary principle to recover detailed information.

Our contributions: Conventional weakly supervised saliency detection network usually start with image-level labels. We claim that scribble annotation is more accurate and also very efficient to obtain. We then introduce the first scribble based saliency detection network [63], and present auxiliary edge detection module together with the structure-aware loss function to achieve effective weakly supervised learning with scribble supervision. Our model greatly reduces the labeling effort and leads to comparable performance compared with the fully supervised models with pixel-wise labeling. Our framework presents a different solution for weakly supervised salient object detection. As we mainly focus on the structure-preserving prediction, the unlabeled pixels fail to contribute effectively to the model updating. More effective solutions can be explored by better utilizing the unlabeled pixels to further improve model performance.

1.2.3 Learning from Noisy Labeling

For the weakly supervised setting, weak annotations are provided as shown in Fig. 1.4. An even cheaper setting is learning directly with noisy labeling, which can be generated with conventional handcrafted feature based methods. Conventionally, the learning from noisy labeling techniques mainly work on image-level classification, and three main directions have been explored: 1) developing regularization techniques [113] to set constraints on the latent clean prediction; 2) estimating the noise distribution [114] by assuming that noisy labels are corrupted from clean labels by an unknown noise transition matrix [114] and 3) training on selected samples [115], which does not require any prior assumption on noise distribution.

Learning image classification from noisy labeling

Zhang and Sabuncu [116] presented a set of noise-robust loss functions, which belongs to the first category. For the second category, Jindal *et al.* [117] proposed a dropout-regularized noise model by augmenting an existing deep network with a noise model that accounts for label noise. Recently, a lot work has been done for the third direction. Jiang *et al.* [118] proposed to learn a MentorNet to produce a curriculum for the StudentNet [119]. The latter one can focus on samples where their labels are possibly correct. Liu and Tao [120] presented an importance reweighting method, where uncertainty introduced by classification noise is reduced by estimating an importance weight parameter. Natarajan *et al.* [121] suggested the use of weighted surrogate loss to reweight the importance of each sample. Ren *et al.* [122] learned to assign weights to training examples based on their gradient directions during each mini-batch. Nguyen *et al.* [123] introduced self-ensemble label filtering to progressively filter out wrong labels during training. Chang *et al.* [115] put more emphasis on uncertain samples to improve mini-batch stochastic gradient-descent (SGD) for image classification.

Learning saliency from noisy labeling
Different from weakly supervised saliency models, where human annotation is still needed, noisy labeling based saliency models can learn saliency without human annotation. The noisy label is computed with conventional handcrafted feature based methods [24]. Zhang *et al.* [124] proposed to fuse noisy saliency maps from handcrafted feature based methods with a heuristic. It combined an intra-image fusion stream and a inter-image fusion stream to generate the learning curriculum and pseudo ground-truth for supervising the training of the deep salient object detector. Nguyen *et al.* [99] defined an image-level loss function to train with multiple noisy labels to generate a coarse saliency map, and then iteratively refine it using a moving average strategy and a fully-connected CRF.
Our contributions: Previous work on noisy labeling based saliency detection [124] usually involves pseudo label generation based on the provided noisy label, which may lead to error propagation due to the less accurate pseudo labels. We provide two alternative solutions: 1) modeling the labeling noise [66, 65] with extra noise estimation module; 2) iteratively identifying pixel-wise noisy annotations [64], and the model is then updated with only loss function based on the clean pixels.

1.3 Thesis Outline

In this thesis, we intend to solve the saliency detection problem in a different way compared with existing techniques via a probabilistic network. Furthermore, we aim to relieve the labeling burden with easy-to-access labels. The thesis is then divided into five parts to address each of these issues.

Part I: Uncertainty Inspired RGB-D Saliency Detection (Chapter 2)

Saliency is subjective to some extent, and different annotators may lead to different attention maps, or saliency maps representing the informative regions for the annotators. In this case, we intend to design a framework to estimate uncertainty for RGB-D saliency detection with a generative model. Particularly, we propose a probabilistic RGB-D saliency detection network via conditional variational autoencoders (CVAE) [51] to model human annotation uncertainty and generate multiple saliency maps for each input image by sampling in the latent space. With the proposed saliency consensus process, we are able to generate an accurate saliency map based on these multiple predictions. Different from existing techniques which learn point estimates to produce a single saliency map for each input image, our algorithms demonstrate the effectiveness of learning the distribution of saliency maps, leading to a new state-of-the-art in RGB-D saliency detection model.

Part II: Complementary Learning for RGB-D Saliency Detection (Chapter 3)

As a multi-modal learning problem, most of the existing RGB-D saliency detec-

tion models [125, 1] focus on implicitly fusing the two modalities at feature-level with different fusion strategies, e.g., early fusion, late fusion or cross-level fusion. We introduce a novel latent variable model based complementary learning framework to explicitly model the complementary information between the two modalities [126]. Specifically, we first design mutual-information minimization as a regularizer to reduce the redundancy between appearance features from RGB and geometric features from depth in the latent space. Then we fuse the latent features of each modality to achieve multi-modal feature fusion. Extensive experiments on benchmark RGB-D saliency datasets illustrate the effectiveness of our framework.

Part III: Probabilistic Coarse-to-Fine Saliency Prediction (Chapter 4)

The latent variable model can model labeling variants, producing uncertainty of prediction, while the energy-based model can evaluate the dependency of each variant, leading to higher-order similarity measure. We then introduce the first probabilistic RGB saliency detection network with an energy-based model whose energy function is parameterized by a bottom-top neural network and a conditional latent variable model whose transformation function is parameterized by an encoder-decoder framework [62]. The conditional latent variable model serves as coarse saliency predictor to generate the initial saliency prediction, and the energy-based model refines the coarse saliency map with an the energy function. The two generative models cooperate to achieve accurate and stochastic saliency prediction. Furthermore, we extend our generative model to weakly supervised learning with a cooperative learning while recovering algorithm. In this way, our model can learn the energy-based model from incomplete data for weakly supervised saliency prediction.

Part IV: Weakly-Supervised Saliency Detection via Scribble (Chapter 5)

Compared with laborious pixel-wise dense labeling, it is much easier to label data by scribbles as shown in Fig. 1.4, which only costs a few seconds to label one image. However, using scribble labels to learn salient object detection has not been explored. We propose a weakly-supervised salient object detection model to learn saliency from such annotations [63]. In doing so, we first relabel an existing large-scale salient object detection dataset with scribbles, namely S-DUTS dataset. Since object structure and detail information is not identified by scribbles, directly training with scribble labels will lead to saliency maps of poor boundary localization. To mitigate this problem, we propose an auxiliary edge detection task to localize object edges explicitly, and a gated structure-aware loss to place constraints on the scope of structure to be recovered.

Part V: Learning Saliency from Noisy Labeling (Chapter 6 and 7)

We define a saliency map computed with conventional handcrafted feature based

methods as a noisy label. With this setting, we present a noise-aware encoder-decoder network [65] to gradually disentangle the latent clean saliency map from the noisy label. With the same setting, we introduce a sampling based principled method [64] to learn saliency from the robust model fitting perspective. Different from [127], where labels are updated according to the outputs of each iteration, we design a strategy and generate an inlier/outlier mask to identify the clean labels. To the best of our knowledge, this is the first time that sampling from single noisy labelling has been used to address the task of saliency prediction.

Chapter 2

Uncertainty Inspired RGB-D Saliency Detection

We propose the first stochastic framework to employ uncertainty for RGB-D saliency detection by learning from the data labeling process. Existing RGB-D saliency detection models treat this task as a point estimation problem by predicting a single saliency map following a deterministic learning pipeline. We argue that, however, the deterministic solution is relatively ill-posed. Inspired by the saliency data labeling process, we propose a generative architecture to achieve probabilistic RGB-D saliency detection which utilizes a latent variable to model the labeling variations. Our framework includes two main models: 1) a generator model, which maps the input image and latent variable to stochastic saliency prediction, and 2) an inference model, which gradually updates the latent variable by sampling it from the true or approximate posterior distribution. The generator model is an encoder-decoder saliency network. To infer the latent variable, we introduce two different solutions: i) a Conditional Variational Auto-encoder with an extra encoder to approximate the posterior distribution of the latent variable; and ii) an Alternating Back-Propagation technique, which directly samples the latent variable from the true posterior distribution. Qualitative and quantitative results on six challenging RGB-D benchmark datasets show our approach's superior performance in learning the distribution of saliency maps. The source code is publicly available via our project page: https://github.com/JingZhang617/UCNet.

2.1 Introduction

Object-level saliency detection (i.e., salient object detection) involves separating the most conspicuous objects that attract human attention from the background [2, 128, 35, 66, 129, 34, 63, 130]. Recently, visual saliency detection from RGB-D images has attracted lots of interests due to the importance of depth information in the human vision system and the popularity of depth sensing technologies [1, 131, 132, 133, 47, 134, 60]. With the extra depth data, conventional RGB-D saliency detection models focus on predicting one single saliency map for the RGB-D input by exploring the complementary information between the RGB image and the depth data.

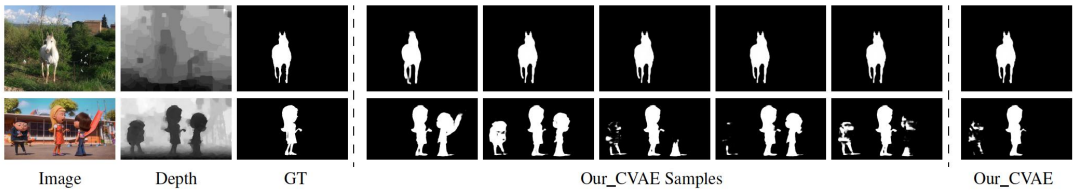


Figure 2.1: GT compared with our predicted saliency maps. For simple context image (first row), we can produce consistent predictions. When dealing with complex scenarios where there exists uncertainties in salient regions (second row), our model can produce diverse predictions ("Our_CVAE Samples"), where "Our_CVAE" is our deterministic prediction after the saliency consensus module, which will be introduced in Section 2.3.3.

The standard practice for RGB-D saliency detection is to train a deep neural network using ground-truth (GT) saliency maps provided by the corresponding benchmark datasets, thus formulating saliency detection as a point estimation problem by learning a mapping function $Y = f(X;\theta)$, where θ represents network parameter set, and X and Y are input RGB-D image pair and corresponding GT saliency map. Usually, the GT saliency maps are obtained through human consensus or by the dataset creators [43]. Building upon large scale RGB-D datasets, deep convolutional neural network-based RGB-D saliency detection models [47, 1, 48, 49, 131] have made profound progress. We argue that the way RGB-D saliency detection progresses through the conventional pipelines [47, 1, 48, 49, 131] fails to capture the uncertainty in labeling the GT saliency maps.

According to research in human visual perception [135], visual saliency detection is subjective to some extent. Each person could have specific preferences [136] in labeling the saliency map (which has been discussed in user-specific saliency detection [137]). More precisely speaking, the GT labeling process is never a deterministic process, which is different from category-aware tasks, such as semantic segmentation [138], as a "Table" will never be ambiguously labeled as "Cat", while the salient foreground for one annotator may be defined as background by other annotators as shown in the second row of Fig. 2.1.

In Fig. 2.1, we present the GT saliency map and other candidate salient regions (produced by our CVAE-based method, which will be introduced in Section 2.3.2) that may attract human attention. Fig. 2.1 shows that the deterministic mapping (from "Image" to "GT") may lead to an "over-confident" model, as the provided "GT" may be biased as shown in the second row of Fig. 2.1. To overcome this, instead of performing point estimation, we are interested in how the network achieves distribution estimation with diverse saliency maps produced[1], capturing the uncertainty of human annotation. Furthermore, in practice, it is more desirable to have multiple saliency maps produced to reflect human uncertainty instead of a single

[1]Diversity of predictions depends on the context of the image, where simple context images will lead to consistent predictions, and complex context images may generate diverse predictions.

saliency map prediction for subsequent tasks.

Inspired by human perceptual uncertainty, as well as the labeling process of saliency maps, we propose a generative architecture to achieve probabilistic RGB-D saliency detection with a latent variable z modeling human uncertainty in the annotation. Two main models are included in this framework: 1) a generator (i.e., encoder-decoder) model, which maps the input RGB-D data and latent variable to stochastic saliency prediction; and 2) an inference model, which progressively refreshes the latent variable. To infer the latent variable, we introduce two different strategies:

- A Conditional Variational Auto-encoder (CVAE) [51] based model with an additional encoder to approximate the posterior distribution of the latent variable.
- The Alternating Back-Propagation (ABP) [139] based technique, which directly samples the latent variable from the true posterior distribution via Langevin Dynamics based Markov chain Monte Carlo (MCMC) sampling [140, 140].

A preliminary version of this work appears in [60], which generates saliency maps via a CVAE and augmented ground-truth to model diversity, and avoids the posterior collapse problem [141]. Although [60] is effective in general, it still has a number of shortcomings. Firstly, [60] requires engineering efforts (ground-truth augmentation) to model diversity and achieve stabilized training (mitigating posterior collapse). As an extension, we use a simpler technique to achieve the same goal, by using the standard KL-annealing strategy [142, 143] with less human intervention. Experimental results in Fig. 2.13 clearly illustrate the effectiveness of the KL-annealing strategy. Secondly, we improve the quality of the generated saliency maps by designing a more expressive decoder that benefits from spatial and channel attention mechanisms [144]. Thirdly, inspired by [51] we modify the cost function of [60] to reduce the discrepancy in encoding the latent variable at training and test time, which is elaborated in Section 2.3.

Moreover, CVAE-based methods approximate the posterior distribution via an inference model (or an encoder) and optimize the evidence lower bound (ELBO). The lower bound is simply the composition of the reconstruction loss and the divergence between the approximate posterior and prior distribution. If the model focuses more on optimizing the reconstruction quality, the latent space may fail to learn meaningful representation. On the other hand, if the model focuses more on reducing the divergence between the approximate posterior and prior distribution, the model may sacrifice the reconstruction quality. Additionally, since the model approximates the posterior distribution rather than modeling the true posterior, it may lose expressivity in general. Here, we propose to use Alternating Back-Propagation (ABP) technique [139] that directly samples latent variables from the true posterior. While it is much simpler, our experimental results show ABP leads to impressive result for generating saliency maps. Note that both CVAE-based and ABP-based solutions can produce stochastic saliency predictions by modeling output space distribution as a generative model conditioned on the input RGB-D image pair. Similar to UC-Net, during the testing phase, a saliency consensus module is introduced to mimic

the majority voting mechanism for GT saliency map generation, and generate one single saliency map in the end for performance evaluation. Finally, in addition to producing state-of-the-art results, our experiments provide a thorough evaluation of the different components of our model as well as an extensive study on the diversity of the generated saliency maps.

Our main contributions are summarized as: 1) We propose the first uncertainty inspired probabilistic RGB-D saliency prediction model with a latent variable z introduced to the network to represent human uncertainty in annotation; 2) We introduce two different schemes to infer the latent variable, including a CVAE [51] framework with an additional encoder to approximate the posterior distribution of z and an ABP [139] pipeline, which samples the latent variable directly from its true posterior distribution via Langevin dynamics based Markov chain Monte Carlo (MCMC) sampling [140]. Each of them can model the conditional distribution of output, and lead to diverse predictions during testing; 3) Extensive experimental results on six RGB-D saliency detection benchmark datasets demonstrate the effectiveness of our proposed solutions.

2.2 Related Work

In this section, we first briefly review existing RGB-D saliency detection models. We then investigate existing generative models, including Variational Auto-encoder (VAE) [50, 51], and Generative Adversarial Networks (GAN) [52, 68]. We also highlight the uniqueness of the proposed solutions in this section.

2.2.1 RGB-D Saliency Detection

Depending on how the complementary information of RGB images and depth data is fused, existing RGB-D saliency detection models can be roughly classified into three categories: early-fusion models [145, 60], late-fusion models [146, 49] and cross-level fusion models [1, 45, 147, 48, 131, 132, 133, 47, 148, 134, 125, 149, 150, 151, 152]. The first solution directly concatenates the RGB image with its depth information, forming a four-channel input, and feed it to the network to obtain both the appearance information and geometric information. [145] proposed an early-fusion model to generate features for each superpixel of the RGB-D pair, which was then fed to a CNN to produce saliency of each superpixel. The second approach treats each modality independently, and predictions from both modalities are fused at the end of the network. [146] introduced a late-fusion network (i.e., , AFNet) to fuse predictions from the RGB and depth branch adaptively. In a similar pipeline, [49] fused the RGB and depth information through fully connected layers. The third one fuses intermediate features of each modality by considering correlations of the above two modalities. To achieve this, [45] presented a complementary-aware fusion block. [48] designed attention-aware cross-level combination blocks to obtain complementary information of each modality. [131] employed a fluid pyramid integration framework to achieve multi-scale cross-modal feature fusion. [133] designed a self-mutual attention model

to effectively fuse RGB and depth information. Similarly, [132] presented a complimentary interaction module (CIM) to select complementary representation from the RGB and depth data. [47] provided joint learning and densely-cooperative fusion framework for complementary feature discovery. [134] introduced a depth distiller to transfer the depth knowledge from the depth stream to the RGB stream to achieve a lightweight architecture without use of depth data at test time. A comprehensive survey can be found in [153].

2.2.2 VAE or CVAE-based Deep Probabilistic Models

Ever since the seminal work by Kingma *et al.* [50] and Rezende *et al.* [154], VAE and its conditional counterpart CVAE [51] have been widely applied in various computer vision problems. A typical VAE-based model consists of an encoder, a decoder, and a loss function. The encoder is a neural network with weights and biases θ, which maps the input datapoint X to a latent (hidden) representation z. The decoder is another neural network with weights and biases ϕ, which reconstructs the datapoint X from z. To train a VAE, a reconstruction loss and a regularizer are needed to penalize the disagreement of the latent representation's prior and posterior distribution. Instead of defining the prior distribution of the latent representation as a standard Gaussian distribution, CVAE-based networks utilize the input observation to modulate the prior on Gaussian latent variables to generate the output.

In low-level vision, VAE and CVAE have been applied to tasks such as latent representations with sharp samples [70], difference of motion modes [71], medical image segmentation models [72], and modeling inherent ambiguities of an image [73]. Further, VAE and CVAE have been explored in more complex vision tasks such as uncertain future forecast [75], salient feature enhancement [74], human motion prediction [76, 155], and shape-guided image generation [77]. Recently, VAE and CVAE have been extended to 3D domain targeting applications such as 3D meshes deformation [78], and point cloud instance segmentation [79]. For saliency detection, [69] adopted VAE to model image background, and separated salient objects from the background through the reconstruction residuals.

2.2.3 GAN or CGAN-based Dense Models

GAN [52] and its conditional counterparts [68] have also been used in dense prediction tasks. Existing GAN-based dense prediction models mainly focus on two directions: 1) using GANs in a fully supervised manner [80, 81, 82, 84, 156] and treat the discriminator loss as a higher-order regularizer for dense prediction; or 2) apply GANs to 'semi-supervised scenarios [85, 86], where the output of the discriminator serves as guidance to evaluate the degree of the unsupervised sample participating in network training. In saliency detection, following the first direction, [83] introduced a discriminator in the fixation prediction network to distinguish predicted fixation map and ground-truth. Different from the above two directions, [157] adopted GAN in a RGB-D saliency detection network to explore the intra-modality (RGB, depth) and

cross-modality simultaneously. [158] used GAN as a denoising technique to clear up the noisy input images. [156] designed a discriminator to distinguish real saliency map (group truth) and fake saliency map (prediction), thus structural information can be learned without CRF [159] as post-processing technique. [160] adopted CycleGAN [161] as an domain adaption technique to generate pseudo-NIR image for existing RGB saliency dataset and achieve multi-spectral image salient object detection.

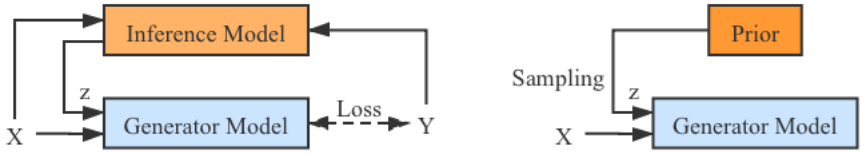
2.2.4 Uniqueness of Our Solutions

To the best of our knowledge, generative models have not been exploited in saliency detection to model annotation uncertainty, except for our preliminary version [60]. As a conditional latent variable model, two different solutions can be used to infer the latent variable. One is CVAE-based [51] method (the one we used in the preliminary version [60]), which infers the latent variable using Variational Inference, and another one is MCMC based method, which we propose to use in this work. Specifically, we present a new latent variable inference solution with less parameter load based on the alternating back-propagation technique [139].

CVAE-based models infer the latent variable through finding the ELBO of the log-likelihood to avoid MCMC as it was too slow in the non-deep-learning era. In other words, CVAEs approximates Maximum Likelihood Estimation (MLE) by finding the ELBO with an extra encoder. The main issue of this strategy is "posterior collapse" [141], where the latent variable is independent of network prediction, making it unable to represent the uncertainty of human annotation. We introduced the "New Label Generation" strategy in our preliminary version [60] as an effective way to avoid posterior collapse problem. In this extended version, we propose a much simpler strategy using the KL annealing strategy[142, 143], which slowly introduces the KL loss term to the loss function with an extra weight. The experimental results show that this simple strategy can avoid the posterior collapse problem with the provided single GT saliency map.

Besides the KL annealing term, we introduce ABP [139] as an alternative solution to prevent posterior collapse in the network. ABP introduces gradient-based MCMC and updates the latent variable with gradient descent back-propagation to directly train the network targeting MLE. Compared with CVAE, ABP samples latent variables directly from its true posterior distribution, making it more accurate in inferring the latent variable. Furthermore, no assistant network (the additional encoder in CVAE) used in ABP, which leads to smaller network parameter load.

We introduce ABP-based inference model as an extension to the CVAE-based pipeline [60]. Experimental results show that both solutions can effectively estimate the latent variable, leading to stochastic saliency predictions. Details of the two inference models are introduced in Section 2.3.2.



(a) Training pipeline (b) Testing pipeline

Figure 2.2: Training and testing pipeline. During training, the inferred latent variable z and input image X are fed to the "Generator Model" for stochastic saliency prediction. During testing, we sample from the prior distribution of z to produce diverse predictions for each input image.

2.3 Our Model

In this section, we present our probabilistic RGB-D saliency detection model, which learns the underlying conditional distribution of saliency maps rather than a mapping function from RGB-D input to a single saliency map. Let $\mathcal{D} = \{X_i, Y_i\}_{i=1}^{N}$ be the training dataset, where X_i denotes the RGB-D input, Y_i denotes the GT saliency map, and N denotes the total number of images in the dataset. We intend to model $P_\omega(Y|X,z)$, where z is a latent variable representing the inherent uncertainty in salient regions which can be also seen in how a human annotates salient objects. Our framework utilizes two main components during training: 1) a generator model, which maps input RGB-D X and latent variable z to conditional prediction $P_\omega(Y|X,z)$; and 2) an inference model, which infers the latent variable z. During testing, we can sample multiple latent variables from the learned prior distribution $P_\theta(z|X)$ to produce stochastic saliency prediction. The whole pipeline of our model during training and testing is illustrated in Fig. 2.2 (a) and (b) respectively. Specifically, during training, the model learns saliency from the "Generator Model", and updates the latent variable with the "Inference Model". During testing, we sample from the "Prior" distribution of the latent variable to obtain stochastic saliency predictions.

2.3.1 Generator Model

The Generator Model takes X and latent variable z as input, and produces stochastic prediction $S = P_\omega(Y|X,z)$, where ω is the parameter set of the generator model. We choose ResNet50 [28] as our backbone, which contains four convolutional blocks. To enlarge the receptive field, we follow DenseASPP [162] to obtain a feature map with the receptive field of the whole image on each stage of the backbone network. We then gradually concatenate the two adjacent feature maps in a top-down manner and feed it to a "Residual Channel Attention" module [163] to obtain stochastic saliency map S. As illustrated in Fig. 2.3, our generator model follows the recent progress in dense prediction problems such as semantic segmentation [138], via a proper use

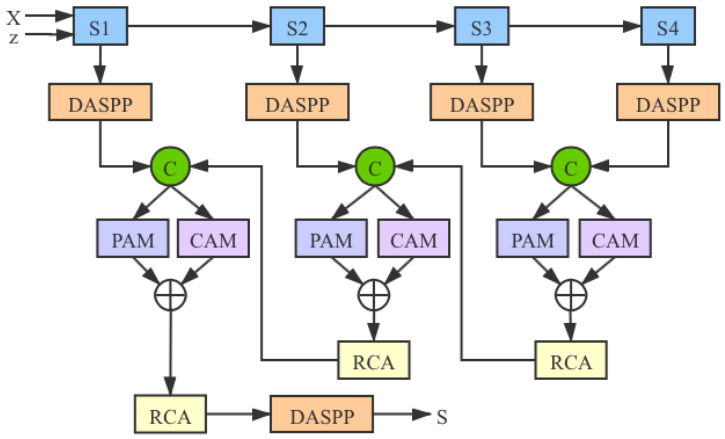


Figure 2.3: Details of the "Generator Model", which takes image X and latent variable z as input, and produce stochastic saliency map S, where "S1-S4" represent the four convolutional blocks of our backbone network. "DASPP" is the DenseASPP module [162], "PAM" and "CAM" are position attention and channel attention module [144], "RCA" is the Residual Channel Attention operation from [163].

of a hybrid attention mechanism. To this end, our generator model benefits from two types of attention: a Position Attention Module [144] and a Channel Attention Module [144]. The former aims to capture the spatial dependencies between any two locations of the feature map, while the latter aims to capture the channel dependencies between any two channel in the feature map. We follow [144] to aggregate and fuse the outputs of these two attention modules to further enhance the feature representations.

2.3.2 Inference Model

We propose two different solutions to infer or update the latent variable z: 1) A CVAE-based [51] pipeline, in which we approximate the posterior distribution via a neural network (i.e., , the encoder); and 2) An ABP [139] based strategy to sample directly from the true posterior distribution of z via Langevin Dynamics based MCMC [140].

Infer z with CVAE: The Variational Auto-encoder [50] is a directed graphical model and typically comprise of two fundamental components, an encoder that maps the input variable X to the latent space $Q_\phi(z|X)$, where z is a low dimensional Gaussian variable and a decoder that reconstructs X from z to get $P_\omega(X|z)$. To train the VAE, a reconstruction loss and a regularizer to penalize the disagreement of the prior and the approximate posterior distribution of z are utilized as:

$$\mathcal{L}_{\text{VAE}} = E_{z\sim Q_\phi(z|X)}[-\log P_\omega(X|z)] + D_{KL}(Q_\phi(z|X)||P(z)), \tag{2.1}$$

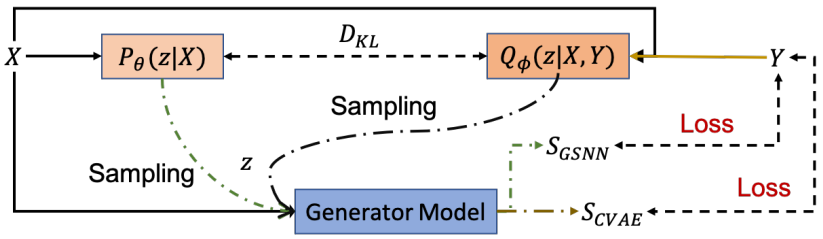


Figure 2.4: RGB-D saliency detection via CAVE. The "Generator Model" is shown in Fig. 2.3. During training, we sample from both posterior net $z \sim Q_\phi(z|X,Y)$ and prior net $z \sim P_\theta(z|X)$ to obtain predictions S_{CVAE} and S_{GSNN} respectively. During testing, S_{GSNN} is our prediction.

where the first term is the reconstruction loss, or the expected negative log-likelihood, and the second term is a regularizer, which is Kullback-Leibler divergence D_{KL} to reduce the gap between the normally distributed prior $P(z)$ and the approximate posterior $Q_\phi(z|X)$. The expectation $E_{z\sim Q_\phi(z|X)}$ is taken with the latent variable z generated from the approximate posterior distribution $Q_\phi(z|X)$.

Different from the VAE, which model marginal likelihood ($P(X)$ in particular) with a latent variable generated from the standard normal distribution, the CVAE [51] modulates the prior of latent variable z as a Gaussian distribution with parameters conditioned on the input data X. There are three types of variables in the conditional generative model: conditioning variable, latent variable, and output variable. In our saliency detection scenario, we define output as the saliency prediction Y, and latent variable as z. As our output Y is conditioned on the input RGB-D data X, we then define the input X as the conditioning variable. For the latent variable z drawn from the Gaussian distribution $P_\theta(z|X)$, the output variable Y is generated from $P_\omega(Y|X,z)$, then the posterior of z is formulated as $Q_\phi(z|X,Y)$, representing feature embedding of the given input-output pair (X,Y).

The loss of CVAE is defined as:

$$\mathcal{L}_{\text{CVAE}} = E_{z\sim Q_\phi(z|X,Y)}[-\log P_\omega(Y|X,z)] + \lambda_{kl} * D_{KL}(Q_\phi(z|X,Y)||P_\theta(z|X)), \quad (2.2)$$

where $P_\omega(Y|X,z)$ is the likelihood of $P(Y)$ given latent variable z and conditioning variable X, the Kullback-Leibler divergence $D_{KL}(Q_\phi(z|X,Y)||P_\theta(z|X))$ works as a regularization loss to reduce the gap between the prior $P_\theta(z|X)$ and the auxiliary posterior $Q_\phi(z|X,Y)$. Furthermore, to prevent the possible *posterior collapse* problem as mentioned in Section 2.2.4, we introduce a linear KL annealing [142, 143] term λ_{kl} as weight for the KL loss term D_{KL}, which is defined as $\lambda_{kl} = ep/N_{ep}$, where ep is current epoch, and N_{ep} is the maximum epoch number. In this way, during training, the CVAE aims to model the conditional log likelihood of prediction under encoding error $D_{KL}(Q_\phi(z|X,Y)||P_\theta(z|X))$. During testing, we can sample from the prior network $P_\theta(z|X)$ to obtain stochastic predictions.

As explained in [51], the conditional auto-encoding of output variables at training

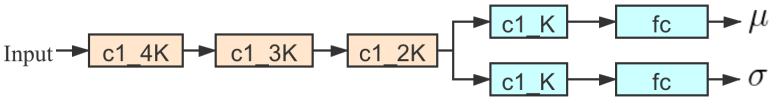


Figure 2.5: Detailed structure of inference models, where K is dimension of the latent space, "c1_4K" represents a 1×1 convolutional layer of output channel size $4 \times K$, "fc" represents the fully connected layer.

may not be optimal to make predictions at test time, as the CVAE uses a posterior of z ($z \sim Q_\phi(z|X,Y)$) for the reconstruction loss in the training stage, while it uses the prior of z ($z \sim P_\theta(z|X)$) during testing. One solution to mitigate the discrepancy in encoding the latent variable at training and testing is to allocate more weights to the KL loss term (e.g., λ_{kl}). Another solution is setting the posterior network the same as the prior network, i.e., , $Q_\phi(z|X,Y) = P_\theta(z|X)$, and we can sample the latent variable z directly from prior network in both training and testing stages. We call this model the "Gaussian Stochastic Neural Network" (GSNN) [51], and the objective function is:

$$\mathcal{L}_{\text{GSNN}} = E_{z\sim P_\theta(z|X)}[-\log P_\omega(Y|X,z)]. \tag{2.3}$$

We can combine the two objective functions introduced above ($\mathcal{L}_{\text{CVAE}}$ and $\mathcal{L}_{\text{GSNN}}$) to obtain a hybrid objective function:

$$\mathcal{L}_{\text{Hybrid}} = \alpha\mathcal{L}_{\text{CVAE}} + (1-\alpha)\mathcal{L}_{\text{GSNN}} \tag{2.4}$$

Following the standard practice of CVAE [51], we design a CVAE-based RGB-D saliency detection pipeline as shown in Fig. 2.4. The two inference models ($Q_\phi(z|X,Y)$ and $P_\theta(z|X)$) share same structure as shown in Fig. 2.5, except for $Q_\phi(z|X,Y)$, we have concatenation of X and Y as input, while $P_\theta(z|X)$ takes X as input. Let's define $P_\theta(z|X)$ as PriorNet, which maps the input RGB-D data X to a low-dimensional latent feature space, where θ is the parameter set of PriorNet. With the provided GT saliency map Y, we define $Q_\phi(z|X,Y)$ as PosteriorNet, with ϕ being the network parameter set. We use five convolutional layers and two fully connected layers to map the input RGB-D image X (or concatenation of X and Y for PosteriorNet) to the statistics of the latent space: $(\mu_{\text{prior}}, \sigma_{\text{prior}})$ for PriorNet and $(\mu_{\text{post}}, \sigma_{\text{post}})$ for PosteriorNet respectively. Then the corresponding latent vector z can be achieved with the reparameteration trick: $z = \mu + \sigma \cdot \epsilon$, where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$.

According to Eq. 2.4, the KL-divergence in $\mathcal{L}_{\text{CVAE}}$ is used to measure the distribution mismatch between the $P_\theta(z|X)$ and $Q_\phi(z|X,Y)$, or how much information is lost when using $Q_\phi(z|X,Y)$ to represent $P_\theta(z|X)$. The GSNN loss term $\mathcal{L}_{\text{GSNN}}$, on the other hand, can mitigate the discrepancy in encoding the latent variable during training and testing. The hybrid loss in Eq. 2.4 can achieve structured outputs with hyper-parameter α to balance the two objective functions in Eq. 2.2 and Eq. 2.3.

Infer z with ABP: As mentioned earlier, one drawback of CVAE-based models is the

Algorithm 1 Learning Stochastic Saliency via Alternating Back-propagation

Input: Training dataset $D = \{(X_i, Y_i)\}_{i=1}^N$
Network Setup: Maximal epoch N_{ep}, number of Langvin steps l, step size s, learning rate γ
Output: Network parameter set ω and the inferred latent variable $\{z_i\}_{i=1}^N$

1: Initialize backbone of the "Generator Model" with ResNet50 [28] for image classification, and other new added layers with a truncated Gaussian distribution. Initialize z_i with standard Gaussian distribution.
2: **for** $t = 1, ..., N_{ep}$ **do**
3: **Inferential back-propagation**: For each i, run l steps of Langevin Dynamics to sample $z_i \sim P_\omega(z_i|Y_i, X_i)$ following Eq. 6.5, with z_i initialized as Gaussian white noise (first iteration) or obtained from previous iteration.
4: **Learning back-propagation**: Update model parameters via: $\omega \leftarrow \omega + \gamma \frac{\partial \mathcal{L}(\omega)}{\partial \omega}$, where the gradient of $\mathcal{L}(\omega)$ can be obtained through stochastic gradient descent.
5: **end for**

posterior collapse problem [141], where the model learns to ignore the latent variable, thus it becomes independent of the prediction Y, as $Q_\phi(z|X, Y)$ will simply collapse to $P_\theta(z|X)$, and z embeds no information about the prediction. In our scenario, the "Posterior Collapse" phenomenon can be interpreted as the fact that the latent variable z fails to capture the inherent human uncertainty in the annotations. To this end, we propose another alternative solution based on alternating back-propagation [139]. Instead of approximating the posterior of z with an encoder network as in a CVAE, we directly sample z from its true posterior distribution via gradient based MCMC.

Alternating Back-Propagation [139] was introduced for learning the generator network model. It updates the latent variable and network parameters in an EM-manner. Firstly, given network prediction with the current parameter set, it infers the latent variable by Langevin dynamics based MCMC, which they call "Inferential back-propagation" [139]. Secondly, given the updated latent variable, the network parameter set is updated with gradient descent, and they call it "Learning back-propagation" [139]. Following the previous variable definitions, given the training example (X, Y), we intend to infer z and learn the network parameter ω to minimize the reconstruction error as well as a regularization term that corresponds to the prior on z.

As a non-linear generalization of factor analysis, the conditional generative model aims to generalize the mapping from continuous latent variable z to the prediction Y conditioned on the input image X. As in traditional factor analysis, we define our generative model as:

$$z \sim P(z) = \mathcal{N}(0, \mathbf{I}), \tag{2.5}$$

$$Y = f_\omega(X, z) + \epsilon, \epsilon \sim \mathcal{N}(0, \text{diag}(\sigma)^2), \tag{2.6}$$

where $P(z)$ is the prior distribution of z. The conditional distribution of Y given X is $P_\omega(Y|X) = \int p(z) P_\omega(Y|X, z) dz$ with the latent variable z integrated out. We define the observed-data log-likelihood as $L(\omega) = \sum_{i=1}^n \log P_\omega(Y_i|X_i)$, where the gradient of

$P_\omega(Y|X)$ is defined as:

$$\frac{\partial}{\partial \omega} \log P_\omega(Y|X) = \frac{1}{P_\omega(Y|X)} \frac{\partial}{\partial \omega} P_\omega(Y|X) = \mathrm{E}_{P_\omega(z|X,Y)} \left[\frac{\partial}{\partial \omega} \log P_\omega(Y, z|X) \right]. \quad (2.7)$$

The expectation term $\mathrm{E}_{P_\omega(z|X,Y)}$ can be approximated by drawing samples from $P_\omega(z|X,Y)$, and then computing the Monte Carlo average. This step corresponds to inferring the latent variable z. Following ABP [139], we use Langevin Dynamics based MCMC (a gradient-based Monte Carlo method) to sample z, which iterates:

$$z_{t+1} = z_t + \frac{s^2}{2} \left[\frac{\partial}{\partial z} \log P_\omega(Y, z_t|X) \right] + s\mathcal{N}(0, I_d), \quad (2.8)$$

with

$$\frac{\partial}{\partial z} \log P_\omega(Y, z|X) = \frac{1}{\sigma^2}(Y - f_\omega(X, z)) \frac{\partial}{\partial z} f_\omega(X, z) - z, \quad (2.9)$$

where t is the time step for Langevin sampling, and s is the step size. The whole pipeline of inferring latent variable z via ABP is shown in Algorithm 1.

Analysis of two inference models: Both the CVAE-based [51] inference model and ABP-based [139] strategy can infer latent variable z, where the former one approximates the posterior distribution of z with an extra encoder, while the latter solution targets at MLE by directly sampling from the true posterior distribution. As mentioned above, the CVAE-based solution may suffer from posterior collapse [141], where the latent variable z is independent of the prediction, making it unable to represent the uncertainty of labeling. In this situation, the latent variable z is absorbed by the network, and the model then produce consistent predictions with multiple iterations of sampling of z from it's prior distribution during testing, leading to a deterministic model instead of a stochastic model. To prevent posterior collapse, we adopt the KL annealing strategy [142, 143], and let the KL loss term in Eq. 2.2 gradually contribute to the CVAE loss function. On the contrary, the ABP-based solution suffers no posterior collapse problem, which leads to simpler and more stable training, where the latent variable z is updated based on the current prediction. In both of our proposed solutions, with the inferred Gaussian random variable z, our model can lead to stochastic prediction, with z representing labeling variants.

2.3.3 Output Estimation

Once the generative model parameters are learned, our model can produce prediction from input X following the generative process of the conditional generative model. With multiple iterations of sampling, we can obtain multiple saliency maps from the same input X. To evaluate performance of the generative network, we need to estimate the deterministic prediction of the structured outputs. Inspired by [51], our first solution is to simply average the multiple predictions. Alternatively, we can obtain multiple z from the prior distribution, and define the deterministic prediction as $Y = f_\omega(X, E(z))$, where $E(z)$ is the mean of the multiple latent variable. Inspired

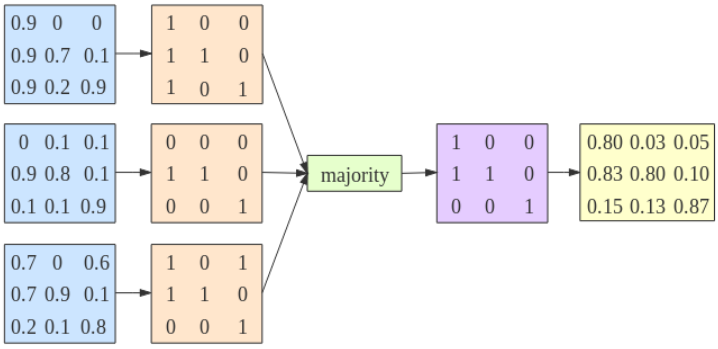


Figure 2.6: Example showing how the saliency consensus module works.

by how the GT saliency map is obtained (e.g., Majority Voting), we introduce a third solution, namely "Saliency Consensus Module", which is introduced in detail.

Saliency Consensus Module: To prepare a training dataset for saliency detection, multiple annotators are asked to label one image, and the majority [43] of saliency regions is defined as being salient in the final GT saliency map.

Although the way in which the GT is acquired is well known in the saliency detection community yet, there exists no research on embedding this mechanism into deep saliency frameworks. The main reason is that current models define saliency detection as a point estimation problem instead of a distribution estimation problem, and the final single saliency map can not be further processed to achieve "majority voting". We, instead, design a stochastic learning pipeline to obtain the conditional distributions of prediction, which makes it possible to perform a similar strategy as preparing the training data to generate deterministic prediction for performance evaluation. Thus, we introduce the saliency consensus module to compute the majority of different predictions in the testing stage as shown in Fig. 2.2 (b).

During testing, we sample z from PriorNet (for the CVAE-based inference model) or directly sample it from a standard Gaussian distribution $\mathcal{N}(0, \mathbf{I})$, and feed it to the "Generator Model" to produce stochastic saliency prediction as shown in Fig. 2.2 (b). With C different samplings, we can obtain C predictions $P^1, ..., P^C$. We simultaneously feed these multiple predictions to the saliency consensus module to obtain the consensus of predictions for performance evaluation.

Given multiple predictions $\{P^c\}_{c=1}^{C}$, where $P^c \in [0, 1]$, we first compute the binary[2] version P_b^c of the predictions by performing adaptive thresholding [164] on P^c. For each pixel (u, v), we obtain a C dimensional feature vector $P_{u,v} \in \{0, 1\}$. We define $P_b^{mjv} \in \{0, 1\}$ as a one-channel saliency map representing the majority of $P_{u,v}$,

[2]As the GT map $Y \in \{0, 1\}$, we produce a series of binary predictions with each one representing annotation from one saliency annotator.

which is defined as:

$$P_b^{mjv}(u,v) = \begin{cases} 1, & \sum_{c=1}^{C} P_b^c(u,v)/C \geq 0.5, \\ 0, & \sum_{c=1}^{C} P_b^c(u,v)/C < 0.5. \end{cases} \tag{2.10}$$

We define an indicator $\mathbf{1}^c(u,v) = \mathbf{1}(P_b^c(u,v) = P_b^{mjv}(u,v))$ representing whether the binary prediction is consistent with the majority of the predictions. If $P_b^c(u,v) = P_b^{mjv}(u,v)$, then $\mathbf{1}^c(u,v) = 1$. Otherwise, $\mathbf{1}^c(u,v) = 0$. We obtain one gray saliency map after saliency consensus as:

$$P_g^{mjv}(u,v) = \frac{\sum_{c=1}^{C}(P_b^c(u,v) \times \mathbf{1}^c(u,v))}{\sum_{c=1}^{C} \mathbf{1}^c(u,v)}. \tag{2.11}$$

We show one toy example with $C = 3$ in Fig. 2.6 to illustrate how the saliency consensus module works. As shown in Fig. 2.6, given three gray-scale predictions (illustrated in blue), we perform adaptive thresholding to obtain three different binary predictions (illustrated in orange). Then we compute a majority matrix (illustrated in purple), which is also binary, with each pixel representing majority prediction of the specific coordinate. Finally, after the saliency consensus module, our final gray-scale prediction is computed based on mean of those pixels agreed (when $P_b^c(u,v) = P_b^{mjv}(u,v)$, we mean in location u, v, the prediction agrees with the majority) with the majority matrix, and ignore others. For example, the majority of saliency in coordinate $(1,1)$ is 1, we obtain the gray prediction after the saliency consensus module as $(0.9 + 0.7)/2 = 0.8$, where 0.9 and 0.7 are predictions in $(1,1)$ of the first and third predictions.

2.3.4 Loss function

We introduce two different inference models to update the latent variable z: a CVAE-based model as shown in Fig. 2.4, and an ABP-based strategy as shown in Algorithm 1. To further highlight structure accuracy of the prediction, we introduce smoothness loss based on the assumption that pixels inside a salient object should have a similar saliency value, and sharp distinction happens along object edges.

As an edge-aware loss, smoothness loss was initially introduced in [165] to encourage disparities to be locally smooth with an L1 penalty on the disparity gradients. It was then adopted in [109] to recover optical flow in the occluded area by using an image prior. We adopt smoothness loss to achieve a saliency map of high intra-class similarity, with consistent saliency prediction inside salient objects, and distinction happens along object edges. Following [109], we define first-order derivatives of the saliency map in the smoothness term as

$$\mathcal{L}_{\text{Smooth}} = \sum_{u,v} \sum_{d \in \overrightarrow{x}, \overrightarrow{y}} \Psi(|\partial_d P_{u,v}| e^{-\alpha |\partial_d Ig(u,v)|}), \tag{2.12}$$

where Ψ is defined as $\Psi(s) = \sqrt{s^2 + 1e^{-6}}$, $P_{u,v}$ is the predicted saliency map at position (u, v), and $Ig(u, v)$ is the image intensity, d indexes over partial derivative in $\overrightarrow{x}$ and $\overrightarrow{y}$ directions. We set $\alpha = 10$ in our experiments following the setting in [109].

We need to compute intensity Ig of the image in the smoothness loss, as shown in Eq. 2.12. To achieve this, we follow a saliency-preserving [166] color image transformation strategy and convert the RGB image I to a gray-scale intensity image Ig as:

$$Ig = 0.2126 \times I^{lr} + 0.7152 \times I^{lg} + 0.0722 \times I^{lb}, \tag{2.13}$$

where I^{lr}, I^{lg}, and I^{lb} represent the color components in the linear color space after Gamma function be removed from the original color space. I^{lr} is achieved via:

$$I^{lr} = \begin{cases} \dfrac{I^r}{12.92}, & I^r \leq 0.04045, \\ \left(\dfrac{I^r + 0.055}{1.055}\right)^{2.4}, & I^r > 0.04045, \end{cases} \tag{2.14}$$

where I^r is the original red channel of image I, and we compute I^g and I^b in the same way as Eq. 2.14.

CVAE Inference Model based Loss Function: For the CVAE-based inference model, we show its loss function in Eq. 2.4, where the negative log-likelihood loss measures the reconstruction error. To preserve structure information and penalize wrong predictions along object boundaries, we adopt the structure-aware loss in [34]. The structure-aware loss is a weighted extension of cross-entropy loss, which integrates the boundary IOU loss [32] to highlight the accuracy of boundary prediction.

With smoothness loss $\mathcal{L}_{\text{Smooth}}$ and CVAE loss $\mathcal{L}_{\text{Hybrid}}$, our final loss function for the CVAE-based framework is defined as:

$$\mathcal{L}_{\text{sal}}^{CVAE} = \mathcal{L}_{\text{Hybrid}} + \lambda_1 \mathcal{L}_{\text{Smooth}}. \tag{2.15}$$

We tested λ_1 in the range of $[0.1, 0.2, \ldots, 0.9, 1.0]$, and found ralatively better performance with $\lambda_1 = 0.3$.

ABP Inference Model based Loss Function: As there exists no extra encoder for the posterior distribution estimation, the loss function for the ABP inference model is simply the negative observed-data log-likelihood:

$$\mathcal{L}_{ABP} = -\sum_{i=1}^{n} \log P_\omega(Y_i | X_i), \tag{2.16}$$

which can be the same structure-aware loss as in [34] similar to CVAE-based inference model.

Integrated with the above smoothness loss, we obtain the loss function for the ABP-based saliency detection model as:

$$\mathcal{L}_{\text{sal}}^{ABP} = \mathcal{L}_{ABP} + \lambda_2 \mathcal{L}_{\text{Smooth}}. \tag{2.17}$$

Similarly, we also empirically set $\lambda_2 = 0.3$ in our experiment.

2.4 Experimental Results

2.4.1 Setup

Datasets: We perform experiments on six datasets including five widely used RGB-D saliency detection datasets (namely NJU2K [167], NLPR [168], SSB [169], LFSD [170], DES [171]) and one newly released dataset (SIP [43]).
Competing Methods: We compare our method with 18 algorithms, including ten handcrafted conventional methods and eight deep RGB-D saliency detection models.
Evaluation Metrics: Four evaluation metrics are used to evaluate the deterministic predictions, including two widely used: 1) Mean Absolute Error (MAE $\mathcal{M}$); 2) mean F-measure (F_β) and two recently proposed: 3) Structure measure (S-measure, S_α) [172] and 4) mean Enhanced alignment measure (E-measure, $E_{\bar{\xi}}$) [173].

- **MAE $\mathcal{M}$:** The MAE estimates the approximation degree between the saliency map Sal and the ground-truth G. It provides a direct estimate of conformity between estimated and GT map. MAE is defined as:

$$\text{MAE} = \frac{1}{N}|Sal - G|, \tag{2.18}$$

  where N is the total number of pixels.

- **S-measure S_α:** Both MAE and F-measure metrics ignore the important structure information evaluation, whereas behavioral vision studies have shown that the human visual system is highly sensitive to structures in scenes [172]. Thus, we additionally include the structure measure (S-measure [172]). The S-measure combines the region-aware (S_r) and object-aware (S_o) structural similarity as their final structure metric:

$$S_\alpha = \alpha * S_o + (1 - \alpha) * S_r, \tag{2.19}$$

  where $\alpha \in [0, 1]$ is a balance parameter and set to 0.5 as default.

- **E-measure $E_{\bar{\xi}}$:** E-measure is the recent proposed Enhanced alignment measure [173] in the binary map evaluation field. This measure is based on cognitive vision studies, which combines local pixel values with the image-level mean value in one term, jointly capturing image-level statistics and local pixel matching information. Here, we introduce it to provide a more comprehensive evaluation.

- **F-measure F_β:** It is essentially a region based similarity metric. We provide the mean F-measure using varying 255 fixed (0-255) thresholds as shown in Fig. 2.7.

Table 2.1: Benchmarking results of seven leading handcrafted feature-based models and eight deep models on six RGBD saliency datasets. ↑ & ↓ denote larger and smaller is better, respectively. Here, we adopt mean F_β and mean E_ξ[173].

		Handcrafted Feature based Models							Deep Models								Ours		
	Metric	GP [174]	CDCP [175]	ACSD [167]	LBE [176]	DCMC [177]	MDSF [178]	SE [179]	DF [145]	AFNet [146]	CTMF [49]	MMCI [147]	PCF [45]	TANet [48]	CPFP [131]	DMRA [1]	UC-Net [60]	CVAE	ABP
NJU2K	S_α ↑	.527	.669	.699	.695	.686	.748	.664	.763	.822	.849	.858	.877	.879	.878	.886	.897	**.902**	.900
	F_β ↑	.357	.595	.512	.606	.556	.628	.583	.653	.827	.779	.793	.840	.841	.850	.873	.886	**.893**	.889
	E_ξ ↑	.466	.706	.594	.655	.619	.677	.624	.700	.867	.846	.851	.895	.895	.910	.920	.930	**.937**	**.937**
	$\mathcal{M}$ ↓	.211	.180	.202	.153	.172	.157	.169	.140	.077	.085	.079	.059	.061	.053	.051	.043	**.039**	**.039**
SSB	S_α ↑	.588	.713	.692	.660	.731	.728	.708	.757	.825	.848	.873	.875	.871	.879	.835	.903	.898	**.904**
	F_β ↑	.405	.638	.478	.501	.590	.527	.611	.617	.806	.758	.813	.818	.828	.841	.837	.884	.878	**.886**
	E_ξ ↑	.508	.751	.592	.601	.655	.614	.664	.692	.872	.841	.873	.887	.893	.911	.879	.938	.935	**.939**
	$\mathcal{M}$ ↓	.182	.149	.200	.250	.148	.176	.143	.141	.075	.086	.068	.064	.060	.051	.066	.039	.039	**.037**
DES	S_α ↑	.636	.709	.728	.703	.707	.741	.741	.752	.770	.863	.848	.842	.858	.872	.900	.934	.937	**.940**
	F_β ↑	.412	.585	.513	.576	.542	.523	.618	.604	.713	.756	.735	.765	.790	.824	.873	.919	**.929**	.928
	E_ξ ↑	.503	.748	.613	.650	.631	.621	.706	.684	.809	.826	.825	.838	.863	.888	.933	.967	**.975**	**.975**
	$\mathcal{M}$ ↓	.168	.115	.169	.208	.111	.122	.090	.093	.068	.055	.065	.049	.046	.038	.030	.019	**.016**	**.016**
NLPR	S_α ↑	.655	.727	.673	.762	.724	.805	.756	.806	.799	.860	.856	.874	.886	.888	.899	**.920**	.917	.919
	F_β ↑	.451	.609	.429	.636	.542	.649	.624	.664	.755	.740	.737	.802	.819	.840	.865	.891	**.893**	.891
	E_ξ ↑	.571	.782	.579	.719	.684	.745	.742	.757	.851	.840	.841	.887	.902	.918	.940	.951	**.952**	**.852**
	$\mathcal{M}$ ↓	.146	.112	.179	.081	.117	.095	.091	.079	.058	.056	.059	.044	.041	.036	.031	.025	.025	**.024**
LFSD	S_α ↑	.640	.717	.734	.736	.753	.700	.698	.791	.738	.796	.787	.794	.801	.828	.847	.864	**.868**	.866
	F_β ↑	.519	.680	.566	.612	.655	.521	.640	.679	.736	.756	.722	.761	.771	.811	.845	.855	.857	**.859**
	E_ξ ↑	.584	.754	.625	.670	.682	.588	.653	.725	.796	.810	.775	.818	.821	.863	.893	.901	**.904**	.903
	$\mathcal{M}$ ↓	.183	.167	.188	.208	.155	.190	.167	.138	.134	.119	.132	.112	.111	.088	.075	.066	**.065**	**.065**
SIP	S_α ↑	.588	.595	.732	.727	.683	.717	.628	.653	.720	.716	.833	.842	.835	.850	.806	.875	**.883**	.876
	F_β ↑	.411	.482	.542	.572	.500	.568	.515	.465	.702	.608	.771	.814	.803	.821	.811	.867	**.877**	.863
	E_ξ ↑	.511	.683	.614	.651	.598	.645	.592	.565	.793	.704	.845	.878	.870	.893	.844	.914	**.927**	.921
	$\mathcal{M}$ ↓	.173	.224	.172	.200	.186	.167	.164	.185	.118	.139	.086	.071	.075	.064	.085	.051	**.045**	.049

Implementation Details: We train our model using PyTorch, and initialized the encoder of the "Generator Model" with ResNet50 [28] parameters pre-trained on ImageNet. Inside the "DASPP" module of the "Generator Model" in Fig. 2.3, we use four different scales of dilation rate: 6, 12, 18, 24 same as [162], and set all intermediate channel size as $M = 32$. For both inference models, we set the dimension of the latent variable as $K = 3$. Weights of new layers are initialized with $\mathcal{N}(0, 0.01)$, and bias is set as constant. We use the Adam method with momentum 0.9 and decrease the learning rate 10% after 80% of the maximum epoch. The base learning rate is initialized as 5e-5. The whole training takes around 9 hours with training batch size 5, and maximum epoch 100 on a PC with an NVIDIA GeForce RTX GPU. For input image size 352×352, the inference time of our CVAE model and ABP model are 0.06s and 0.05s on average respectively.

2.4.2 Comparison to State-of-the-art Methods

Quantitative Comparison: We report the performance of our method (with both inference models) and competing methods in Table 2.1, where "CVAE" is our framework with CVAE as inference model, and "ABP" represents the model that updates latent variable z with alternating back-propagation. Results in Table 2.1 demonstrate

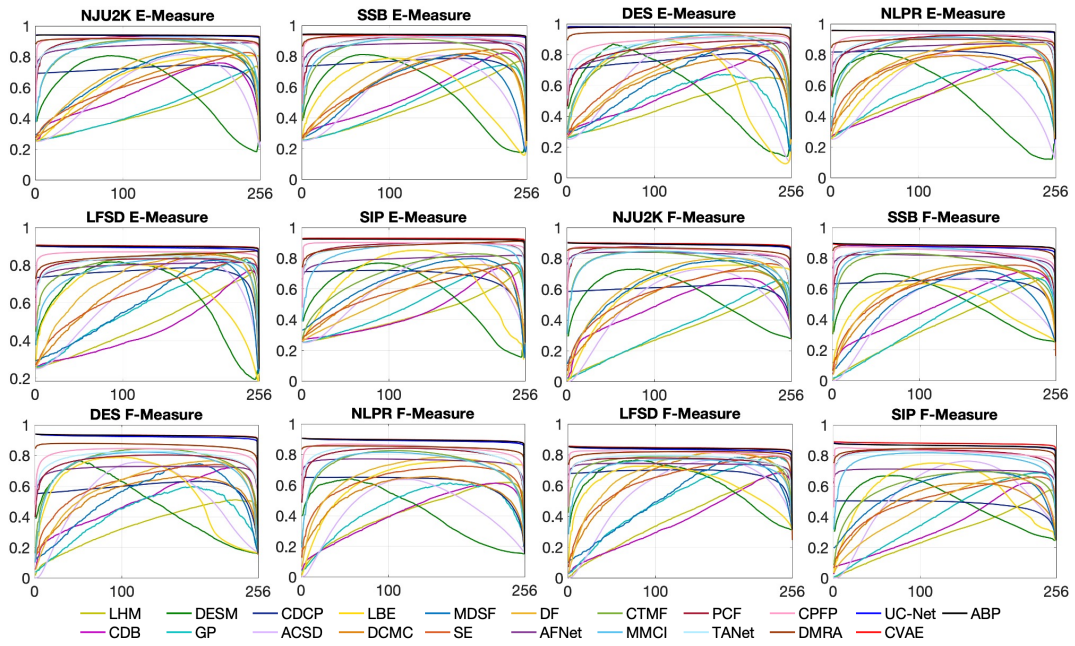


Figure 2.7: E-measure and F-measure curves on six testing datasets (NJU2K, SSB, DES, NLPR, LFSD and SIP). Best viewed on screen.

the benefits of both CVAE and ABP which consistently achieve the best performance on all datasets. Specifically, on SSB [169] and SIP [43], our method achieves around a 2.5% S-measure, E-measure and F-measure performance boost and a decrease in MAE by 1.5% compared with the "Deep Models" in Table 2.1. Moreover, compared with our preliminary version "UC-Net" [60], we observe improved performance, which indicates the effectiveness of the proposed structure. We also show E-measure and F-measure curves of competing methods and ours in Fig. 2.7. We observe that our method produces not only stable E-measure and F-measure but also the best performance.

To further evaluate the proposed method, we compute performance of eight cutting-edge RGB saliency detection models on the RGB-D testing dataset[3] and compared with our "CVAE" based model. The results are shown in Table 2.2, which further illustrates the superior performance of the proposed framework.

Qualitative Comparisons: In Fig. 2.8, we show five examples comparing our method with six RGB-D saliency detection models. Salient objects in these images can be large (fifth row), small (second row) or in complex backgrounds (first, third, fourth and fifth rows). Especially for the example in the first row, the background is complex, part of the background shares similar color and texture as the salient foreground. Most of those competing methods (AFNet[146], CPFP[131] and DMRA[1])

[3]The RGB saliency models are trained on RGB saliency training set, and testing on RGB-D testing set, where the depth is not used.

Table 2.2: Performance of competing RGB saliency detection models and ours on RGBD saliency datasets, where depth data is not used while testing using the RGB saliency models. We adopt mean F_β and mean E_ξ.

	Metric	AFBNet [40]	NLDF [32]	PiCANet [31]	RAS [180]	DGRL [181]	CPD [37]	SCRN [130]	F3Net [34]	CAVE Ours
NJU2K[167]	$S_\alpha \uparrow$	.862	.813	.864	.754	.767	.875	.879	.861	**.902**
	$F_\beta \uparrow$	.835	.783	.818	.744	.716	.852	.863	.837	**.893**
	$E_\xi \uparrow$	.888	.848	.869	.800	.804	.903	.912	.890	**.937**
	$\mathcal{M} \downarrow$	.064	.091	.072	.115	.107	.056	.052	.061	**.039**
SSB[169]	$S_\alpha \uparrow$	.893	.859	.896	.828	.824	.902	.902	.891	**.898**
	$F_\beta \uparrow$	.865	.831	.844	.820	.781	.880	.881	.868	**.878**
	$E_\xi \uparrow$	.918	.893	.899	.871	.865	.928	.928	.921	**.935**
	$\mathcal{M} \downarrow$	.045	.062	.053	.076	.073	.040	.041	.043	**.039**
DES[171]	$S_\alpha \uparrow$	.879	.828	.883	.806	.833	.894	.907	.880	**.937**
	$F_\beta \uparrow$	.845	.758	.822	.762	.753	.870	.885	.845	**.929**
	$E_\xi \uparrow$	.893	.831	.872	.823	.849	.907	.927	.892	**.975**
	$\mathcal{M} \downarrow$	.035	.058	.039	.060	.054	.029	.026	.030	**.016**
NLPR[168]	$S_\alpha \uparrow$	.881	.847	.876	.853	.840	.893	.894	.884	**.917**
	$F_\beta \uparrow$	.816	.782	.789	.810	.767	.844	.846	.838	**.893**
	$E_\xi \uparrow$	.896	.876	.870	.888	.873	.914	.920	.912	**.952**
	$\mathcal{M} \downarrow$	.042	.052	.051	.049	.053	.034	.036	.035	**.025**
LFSD[170]	$S_\alpha \uparrow$	.817	.777	.827	.673	.782	.836	.827	.835	**.868**
	$F_\beta \uparrow$	.784	.756	.778	.672	.759	.811	.800	.810	**.857**
	$E_\xi \uparrow$	.838	.806	.825	.727	.817	.856	.847	.857	**.904**
	$\mathcal{M} \downarrow$	.094	.121	.103	.162	.117	.088	.088	.089	**.065**
SIP[43]	$S_\alpha \uparrow$	.876	.795	.851	.718	.682	.870	.866	.866	**.883**
	$F_\beta \uparrow$	.847	.752	.806	.696	.606	.859	.861	.850	**.877**
	$E_\xi \uparrow$	.911	.840	.866	.766	.744	.910	.903	.905	**.927**
	$\mathcal{M} \downarrow$	.055	.100	.073	.121	.138	.053	.057	.055	**.045**

failed to correctly segment the precise salient foreground, while our approach achieves better salient object detection with each of the proposed two inference models. For the image in the last row, there exists an object (i.e., , green toy) that strongly stands out from its background, while the depth map can to some extent decrease the salience of such high-contrast region. All of the competing methods (DCMC[177], SE[179], AFNet[146], CPFP[131] in particular) falsely detect part of the background region as being salient, whereas our accurate predictions further indicate the effectiveness of our solutions. With all the results in Fig. 2.8, we can see evidence of the superiority of our approach.

Probabilistic Distribution Evaluation: As a probabilistic network, our models can produce a distribution of plausible saliency maps instead of a single, deterministic prediction for each input image. We argue that, for images with simple background, consistent predictions should be produced, whereas for complex images with cluttered background, we expect our model to capture the uncertainty in the saliency maps, and thus can generate diverse predictions. To evaluate performance of our model, following the active learning pipeline [182], we first generate $B = 100$ easy and difficult samples. To achieve this, we first adopt three different conventional saliency models (RBD [24], MR [26] and GS [183], which rank among the top

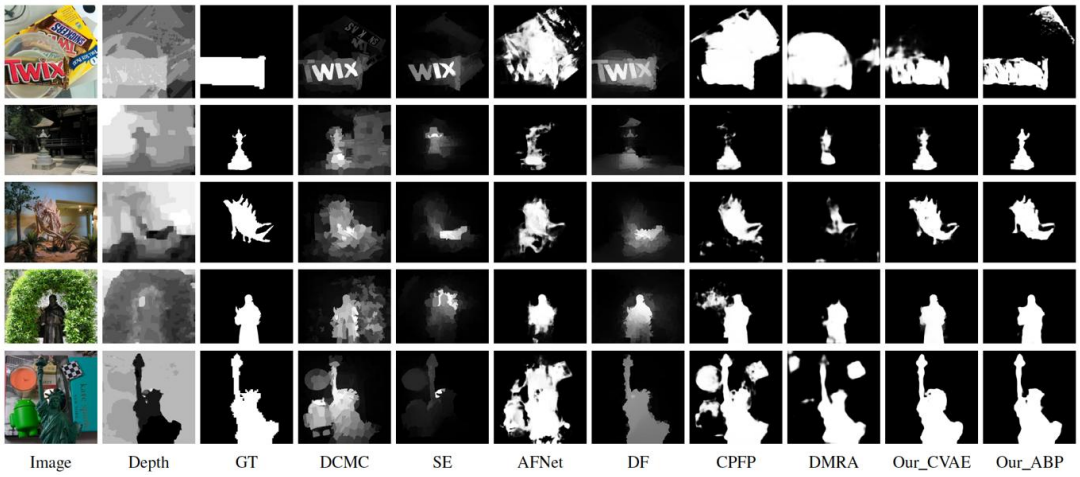


Figure 2.8: Visual comparison of predictions of our methods and competing methods. Note that, our final prediction is generated with the proposed "Saliency Consencus Module" (see Section 2.3.3).

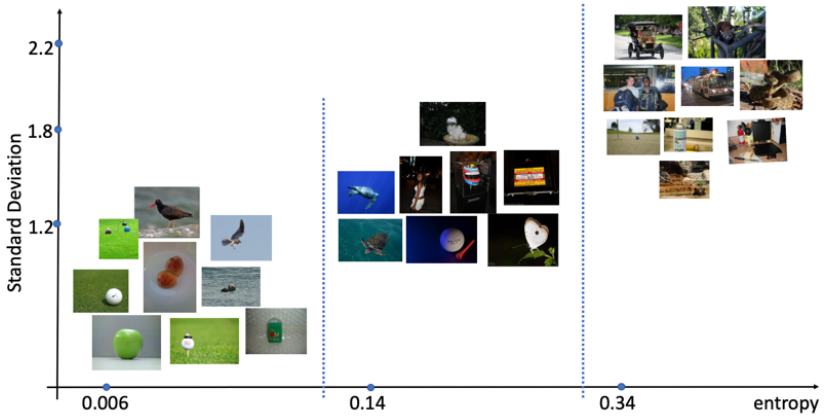


Figure 2.9: Image distribution by analysing entropy and standard deviation.

six conventional handcrafted feature based RGB saliency models [164]), and define them as $f1$, $f2$ and $f3$ respectively. Given image X_i[4] in training dataset D, we compute its corresponding saliency map $f1(X_i)$, $f2(X_i)$ and $f3(X_i)$. We choose entropy as measure for image complexity. Then, we define mean saliency map of X_i as $P_i = (f1(X_i) + f2(X_i) + f3(X_i))/3$. We define the complexity of the image as task driven (for saliency detection). Then given a ground-truth saliency map Y_i and mean saliency map P_i, we define foreground entropy as: $-P_i \log P_i$.

We then define mean entropy as a complexity measure, and choose B images with the smallest entropy as the easy samples and B images with the largest entropy as the

[4]We use the RGB data only.

Table 2.3: The code type and inference time of existing approaches. M = Matlab. Pt = PyTorch. Tf = Tensorflow.

Method	LHM [168]	CDB [184]	DESM [171]	GP [174]	CDCP [175]	ACSD [167]	LBE [176]	DCMC [177]	MDSF [178]
Time (s)	2.13	0.60	7.79	12.98	60.00	0.72	3.11	1.20	60.00
Code Type	M	M	M	M&C++	M&C++	C++	M&C++	M	C++
Method	SE [179]	DF [145]	AFNet [146]	CTMF [49]	MMCI [147]	PCF [45]	CPFP [131]	Our_ABP	Our_CVAE
Time (s)	1.57	10.36	0.03	0.63	0.05	0.06	0.17	0.05	0.06
Code Type	M&C++	M&C++	Tf	Caffe	Caffe	Caffe	Caffe	Pt	Pt

difficult samples (with $B = 100$). We sample $Sn = 5$ times from the prior distribution and compute the variance of each group. Specifically, for image pair X_i, with Sn iterations of sampling, we obtain its prediction $\{S_i^j\}_{j=1}^{Sn}$. We compute the similarity of these Sn different predictions, and treat it as prediction diversity evaluation. We show entropy and standard deviation of images in Fig. 2.9.

Inference Time[5] and Model Complexity Comparison: We summarize basic information of competing methods in Table 2.3 for clear comparison, including their code type and inference time. Table 2.3 shows that the inference time[6] of our method is comparable with competing methods, which further illustrates that our model can achieve probabilistic predictions with no inference time sacrificed. Further, the existing RGB-D saliency detection models (AFNet [146], CTMF [49], MMCI [147], PCF [45], CPFP [131]) adopt the VGG16 network [67] as backbone to achieve cross-level feature fusion, where there exists two sets of VGG16 [67] backbones for both RGB image and depth feature extraction. We use the ResNet50 backbone [28]. Although ResNet50 backbone has more parameters (around 50M) than the VGG16 backbone (around 30M), as an early fusion model, we use one copy of the backbone network for feature extraction, leading to comparable a parameters number compared with existing techniques.

2.4.3 Structured Output Generation

As a generative network, we introduce a latent variable z modeling uncertainty of human annotation. We further show examples of our model generating structured outputs as shown in Fig. 2.10. The "Our_CVAE Samples" in Fig. 2.10 represents three random samples of our method with the CVAE inference model, and "Our_ABP Samples" are samples with the ABP strategy. "Our_CVAE" and "Our_ABP" are the deterministic predictions of our frameworks with the above two inference models obtained via our "Saliency Consensus Module". Fig. 2.10 shows that both the two inference models can produce reasonable stochastic predictions, and the final deterministic prediction after the "Saliency Consensus Module" ("Our_CVAE" and

[5]Conventional handcrafted-feature based methods are implemented on CPU, and deep RGB-D saliency prediction models are based on GPU, thus we report CPU time for the former and GPU time for the later.

[6]The inference time we report represents prediction with one random sampling from the PriorNet.

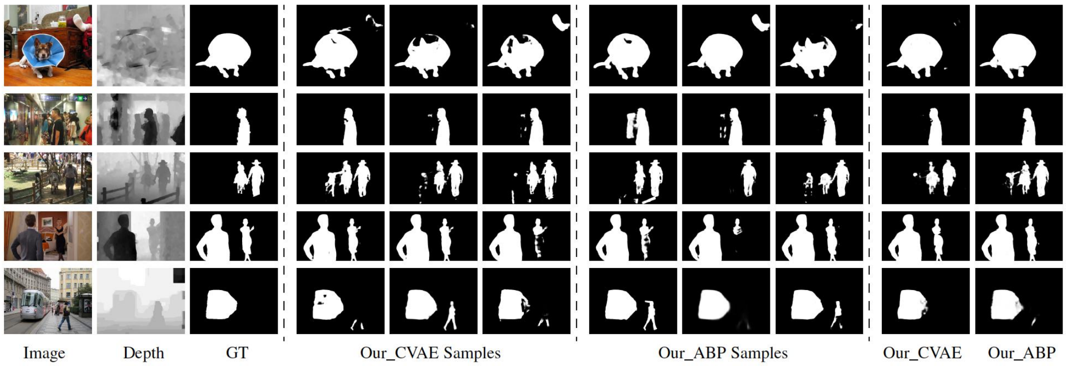
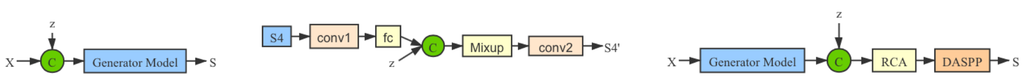


Figure 2.10: Structured outputs generation, where “Our_CVAE Samples” and “Our_CVAE” are samples and the deterministic prediction respectively.

Table 2.4: Evaluation of the effect of different components in our models, and alternative structures. We present mean F_β and mean E_ξ.

	NJU2K				SSB				DES				NLPR				LFSD				SIP			
Method	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$
Middle	.897	.888	.933	.042	.895	.880	.934	.041	.931	.920	.968	.018	.916	.887	.950	.026	.854	.843	.888	.073	.873	.863	.914	.048
Late	.890	.875	.929	.046	.891	.866	.931	.042	.929	.909	.970	.020	.907	.877	.947	.028	.839	.828	.887	.076	.870	.853	.916	.051
AveP	.900	.892	.936	.040	.897	.877	.934	.040	.935	.924	.970	.017	.914	.890	.951	.025	.857	.842	.899	.067	.880	.876	.926	.046
AveZ	.901	.890	.927	.040	.892	.875	.930	.040	.929	.921	.971	.018	.914	.884	.950	.026	.855	.843	.892	.068	.880	.874	.926	.046
GSNN	.900	.887	.935	.040	.894	.873	.930	.041	.931	.919	.971	.018	.913	.885	.949	.026	.852	.834	.894	.070	.871	.864	.916	.051
CVAE_S	.900	.890	.932	.040	.894	.876	.931	.041	.936	.927	.974	.016	.914	.891	.949	.026	.856	.843	.897	.068	.877	.867	.920	.048
NoS	.893	.881	.933	.042	.885	.876	.930	.044	.931	.921	.966	.017	.914	.878	.950	.027	.853	.845	.898	.069	.882	.868	.924	.047
CE	.900	.891	.936	.041	.894	.876	.930	.040	.935	.921	.970	.018	.913	.891	.950	.025	.851	.833	.887	.075	.876	.856	.916	.051
HHA	.897	.886	.934	.042	.902	.882	.937	.038	.930	.917	.970	.019	**.919**	.892	.950	.024	.850	.834	.888	.074	.870	.856	.915	.052
w/o KLA	.900	.890	.932	.041	.893	.870	.931	.040	.932	.923	.972	.017	.913	.887	.948	.027	.854	.841	.893	.069	.881	.872	.923	.046
Our_CAVE	**.902**	**.893**	**.937**	**.039**	.898	.878	.935	.039	.937	**.929**	**.975**	**.016**	.917	**.893**	**.952**	.025	**.868**	.857	**.904**	**.065**	**.883**	**.877**	**.927**	**.045**
Our_ABP	.900	.889	**.937**	**.039**	**.904**	**.886**	**.939**	**.037**	**.940**	.928	**.975**	**.016**	**.919**	.891	**.852**	**.024**	.866	**.859**	.903	**.065**	.876	.863	.921	.049



(a) Early fusion model (b) Middle fusion model (c) Late fusion model

Figure 2.11: Detail network structures of different fusion schemes: the early fusion model (a), the middle fusion model (b) and the late fusion model (c).

“Our_ABP”) is consistent with the provided GT, which verifies effectiveness of both our latent variable and the “Saliency Consensus Module”.

2.4.4 Ablation Studies

We further analyse the proposed framework in this section, including the generative network related strategies, the loss functions, the alternative depth data (HHA [185] in particular), and the solution to prevent network from posterior collapse. We show the performance in Table 2.4. Note that unless otherwise stated, we use the CVAE-based inference model in the following experiments.

Different Fusion Schemes: The latent variable z can be fused to the network in three different ways: early fusion (in the input layer), middle fusion (in bottleneck network), or late fusion (before the output layer). We propose an early fusion model as shown in Fig. 2.11 (a). We further design a middle fusion models and a late fusion model as shown in Fig. 2.11 (b) and (c) respectively. The performance of each model is shown in Table 2.4 "Middle" and "Late". For the middle fusion model, last convolutional layer of the fourth group (e.g., S4) of the backbone network is fed to a 1×1 convolutional layer to obtain a $M = 32$ dimensional feature map, which is then map to a K (dimension of the latent variable z) dimensional feature vector with a fully connected layer ("fc"). To avoid posterior collapse [141], inspired by [76], we mix ("Mixup") the feature vector and z channel-wise; thus, the network cannot distinguish between features of the deterministic branch and the probabilistic branch. We then expand the mixed feature vector in the spatial dimension, and feed it to another 1×1 convolutional layer to achieve feature map S4' of the same dimension as S4, and replace S4 with S4' in Fig. 2.3. For the late fusion model, the "Generator Model" represents the generator model in Fig. 2.3 before the last "RCA" module. We expand z in spatial dimension and concatenate it with the deterministic feature. We also perform "Mixup" here similar to the middle fusion model. We then feed the mixed feature map to one "RCA" module and "DASPP" model to achieve prediction S. We observe slightly worse performance of the middle fusion model ("Middle") and late fusion model ("Late"). The main reason is that strong non-linear representation can be obtained when the latent variable is fed to the beginning of the network, which is also consistent with the result that "Middle" is better than "Late".

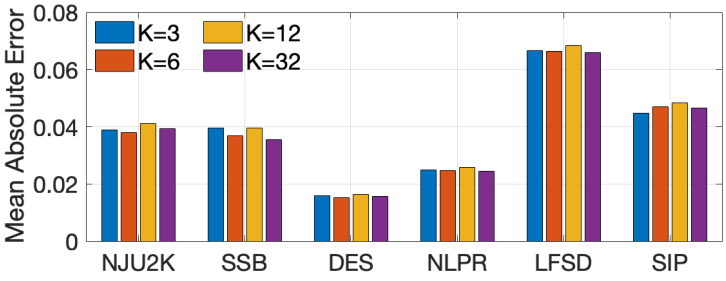


Figure 2.12: Dimension analysis of the latent variable.

Analysing the Effect of the Dimension of z: The scale of z may influence both network performance and diversity of predictions. In our experiments, we set dimension of z to 3. We further carry out experiments with dimension of z in the range

of [3, 32], and show mean absolution error of our model on six benchmark RGB-D saliency dataset in Fig. 2.12. We observe relatively stable performance for different dimension of z. The relatively stable performance regardless of dimension of z shows that the capacity of the network is large enough to take different degree of stochasticity in the input. Further, as there exists only a few quite difficult samples, and lower dimension of z is enough to capture variants of labeling.

Deterministic Prediction Generation: As introduced in Section 2.3.3, three different solutions can be used to generate a deterministic prediction for performance evaluation, including 1) averaging multiple predictions; 2) averaging multiple latent variables; and 3) the proposed saliency consensus module. We evaluate performance of other deterministic inference solutions and show performance in Table 2.4 "AveP" and "AveZ", representing the average-prediction solution and average-z solution respectively. We observe similar performance of "AveP" and "AveZ" compared with the proposed saliency consensus module. The similar performance of "AveP" and "AveZ" illustrates that both conventional deterministic prediction generation solutions work well for the saliency detection task. The better performance of "Ours" indicates effectiveness of the proposed solution.

Effectiveness of Loss Functions: Due to the inconsistency of $Q_\phi(z|X,Y)$ and $P_\theta(z|X)$ used in the training and testing stage respectively, the model may behave differently during training and testing. To mitigate the discrepancy in encoding the latent variable, and achieve similar network behavior during training and testing, we introduce Gaussian Stochastic Neural Network (GSNN) and a hybrid loss function as shown in Eq. 2.4. To test how our network performs with only the CVAE loss in Eq. 2.2 or GSNN loss in Eq. 2.3, we train two extra models and show performance as "CVAE_S" and "GSNN" respectively. We see clear performance decreased with each loss used solely. Further, although the two models perform worse than the proposed solution, we still observe consistent better performance compared with competing methods. Both the performance drop of "CVAE_S" and "GSNN" compared with "Ours", and better performance of "CVAE_S" and "GSNN" compared with competing methods, indicate effectiveness of the proposed generative model for saliency detection.

Smoothness Loss: We introduce the smoothness loss to our loss function to set constraints on the structure of the prediction. To evaluate the contribution of the smoothness loss, we remove it from our loss function and show the performance as "NoS". The lower performance indicates the effectiveness of the smoothness loss. Moreover, as shown in Eq. 2.12, the smoothness loss takes saliency prediction and gray-scale image as input, which can also be interpreted as a self-supervised regularizer.

Structure-aware Loss *vs*. **Cross-entropy Loss:** Similar to [34], we use structure-aware loss instead of the widely used cross-entropy loss to penalize prediction along object edges, thus we can achieve structure-preserving saliency prediction. To prove that our model can also works well with basic cross-entropy loss, we designed another model with cross-entropy loss used instead of the structure-aware loss, and show performance as "CE". We notice clear decreased performance of "CE" on "LFSD" and "SIP" dataset. For both "LFSD" and "SIP" dataset, there exists salient foreground regions that share similar color as the background, which makes the cross-entropy

Table 2.5: Comparison with the state-of-the-art RGB saliency detection models on six benchmark RGB saliency datasets. We adopt mean F_β and mean E_ξ.

	DUTS				ECSSD				DUT				HKU-IS				THUR				SOC			
Method	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$
DGRL [181]	.846	.790	.887	.051	.902	.898	.934	.045	.809	.726	.845	.063	.897	.884	.939	.037	.816	.727	.838	.077	-	-	-	-
PiCAN [31]	.842	.757	.853	.062	.898	.872	.909	.054	.817	.711	.823	.072	.895	.854	.910	.046	.818	.710	.821	.084	.801	.332	.810	.133
NLDF [32]	.816	.757	.851	.065	.870	.871	.896	.066	.770	.683	.798	.080	.879	.871	.914	.048	.801	.711	.827	.081	.816	.319	.837	.106
BASN [33]	.876	.823	.896	.048	.910	.913	.938	.040	.836	.767	.865	.057	.909	.903	.943	.032	.823	.737	.841	.073	.841	.359	.864	.092
AFNet [40]	.867	.812	.893	.046	.907	.901	.929	.045	.826	.743	.846	.057	.905	.888	.934	.036	.825	.733	.840	.072	.700	.062	.684	.115
MSNet [39]	.862	.792	.883	.049	.905	.886	.922	.048	.809	.710	.831	.064	.907	.878	.930	.039	.819	.718	.829	.079	-	-	-	-
SCRN [130]	.885	.833	.900	.040	.920	.910	.933	.041	.837	.749	.847	.056	.916	.894	.935	.034	.845	.758	.858	.066	.838	.363	.859	.099
LDF [188]	**.890**	.861	.925	**.034**	.919	.923	.943	.036	.839	.770	.865	.052	.920	.913	.953	.028	.842	.768	.863	**.064**	-	-	-	-
Ours_CVAE	.888	.860	.927	**.034**	**.921**	**.926**	**.947**	**.035**	.839	**.773**	**.869**	.051	**.921**	**.919**	**.957**	**.026**	.848	.765	.862	**.064**	**.849**	**.369**	**.872**	**.089**
Ours_ABP	**.890**	**.864**	**.931**	**.034**	.915	.918	.941	.037	**.843**	.770	.864	**.050**	.917	.913	.949	.027	**.849**	**.773**	**.869**	.066	.842	.365	.868	.091

based model ineffective in those scenarios. While the structure-aware loss can penalize prediction with wrong structure information, making it effective for those difficult images.

HHA *vs*. **Depth:** HHA [185] is a widely used technique that encodes the depth data to three channels: **h**orizontal disparity, **h**eight above ground, and the **a**ngle the pixel's local surface normal makes with the inferred gravity direction. HHA is widely used in RGB-D dense models [186, 49] to obtain better feature representation. To test if HHA also works in our scenario, we replace depth with HHA, and performance is shown in "HHA". We observe similar performance achieved with HHA instead of the raw depth data. Those models using HHA aim to obtain better depth representation, as the raw depth is not usually in low-quality. The proposed stochastic model introduces randomness to the network, which can also serve as denoising technique to improve robustness of the model, and this is also consistent with the observation in [187].

Training without KL Annealing: As discussed in Section 2.2.4, we introduce KL annealing strategy to prevent the possible posterior collapse problems of the CVAE-based model. To test contribution of this strategy, we simply remove the KL annealing term, and set weight of the KL loss term in Eq. 2.2 as 1 from the first epoch. Performance of this experiment is shown as "w/o KLA". Although the performance on the six benchmark RGB-D saliency datasets does not show effect of KL annealing clearly (as we generate a deterministic prediction), we observed that it highly affects the diversity of the prediction as shown in Fig. 2.13, which presents the mean variance of multiple predictions on the RGB-D testing sets. Specifically, we perform five iterations of random sampling during testing, and compute variance of those five different predictions. We show mean of the variance maps in Fig. 2.13. Further, we show the mean variance of our CVAE-based and ABP-based models as "CVAE" and "ABP" respectively. Fig. 2.13 clearly shows that both of our proposed solutions can generate more diverse predictions than "w/o KLA", leading to larger variance than "w/o KLA".

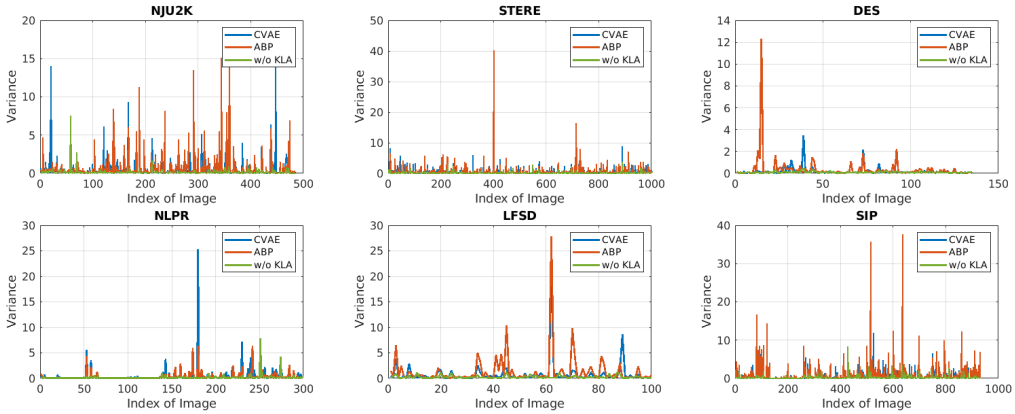


Figure 2.13: Mean variance of multiple predictions using our CAVE-based model ("CVAE"), ABP-based model ("ABP"), and the CAVE-based model without KL annealing term ("w/o KLA"). Best viewed on screen.

2.4.5 Probabilistic RGB Saliency Detection

We propose a generative model based RGB-D saliency detection network, and we extend it to RGB saliency detection to test flexibility of the proposed framework, and show performance in Table 2.5. We train our model ("Ours_CVAE" and "Ours_ABP") with DUTS training dataset [41], and evaluate performance of our methods and competing methods on six widely-used benchmarks: (1) DUTS testing dataset; (2) ECSSD [189]; (3) DUT [26]; (4) HKU-IS [30]; (5) THUR [190] and (6) SOC [191]. Note that, similar to the RGB-D based framework, we use the same network structure, except that the input image X is RGB data instead of the RGB-D image pair. The consistent better performance of our network ("Ours_CVAE" or "Ours_ABP") illustrates flexibility of our model, which can be lead to new benchmark performance for both RGB-D saliency detection and RGB saliency detection.

2.5 Conclusion

Inspired by human uncertainty in ground-truth annotation, we proposed the first uncertainty inspired RGB-D saliency detection model. Different from existing methods, which generally treat saliency detection as a point estimation problem, we propose to learn the distribution of saliency maps, and proposed a generative learning pipeline to produce stochastic saliency predictions. Further, we introduce two different inference models: 1) a CVAE-based inference model, where an extra encoder to approximate true posterior distribution of the latent variable z; and 2) an ABP-based inference model to sample z directly from its true posterior distribution with gradient based MCMC. Under our formulation, our model is able to generate multiple predictions, representing uncertainty of human annotation. With the proposed saliency consensus module, we are able to produce accurate saliency prediction following a similar pipeline to the ground-truth annotation generation process. Quantitative and

qualitative evaluations on six standard and challenging benchmark RGB-D datasets demonstrated the superiority of our approach in learning the distribution of saliency maps.

Furthermore, we thoroughly investigate the generative model and include analysis of both the latent variable, the loss function and the different fusion schemes to introduce z to the network. We observe that our model is mostly influenced by the dimension of the latent variable, which not only has impact on deterministic prediction performance, but also influences the stochastic predictions. We will investigate the effectiveness of the latent variable with uncertainty estimation techniques [192]. As an extension, we also perform RGB saliency detection. Without changing network structure (we only change the input from RGB-D data to RGB data), we achieve state-of-the-art performance compared with the last RGB saliency models, which further explains superiorty of our solution.

Two different inference models are introduced to learn the proposed generative network as shown in Fig. 2.2 (a). From our experience, both the CVAE-based and ABP-based inference models can lead to diverse saliency predictions as shown in Fig. 2.13. However, as extra encoder is used in the CVAE model, it leads to more network parameters than the ABP-based solution. On the other hand, as we update the latent variable by running multiple steps of Langevin Dynamics based MCMC as shown in Eq. 2.8, which leads to relatively longer training time for the ABP-based solution. Combining the advantage of both techniques to achieve both efficient training and effective stochastic predictions is our next step.

Chapter 3

RGB-D Saliency Detection via Cascaded Mutual Information Minimization

Existing RGB-D saliency detection models do not explicitly constrain RGB and depth modes to achieve effective cooperative learning. In Chapter 2, we introduced the uncertainty-aware RGB-D saliency detection framework [60], which is an early-fusion model that directly fuses the two modes in the input layer, thus it fails to explicitly model the relationship between the RGB image and depth data for saliency detection. In this chapter, we introduce a novel latent variable model based complementary learning framework to **explicitly** model the complementary information between the two modes. Specifically, we first design mutual-information minimization as a regularizer to reduce the redundancy between appearance features from RGB and geometric features from depth in the latent space. Then we fuse the latent features of each mode to achieve multi-mode feature fusion. Extensive experiments on benchmark RGB-D saliency datasets illustrate the effectiveness of our framework. To prosper the development of this field, we further contribute the largest (10$\times$scale of previous) Ours dataset, which contains 20,625 image pairs with 15,625 high quality polygon-/scribble-/object-/instance-/rank-level annotations. Based on these rich labels, we additionally conduct three new benchmarks[1] with strong baselines and observe some interesting phenomena, which can help motivate future model design. We believe our systematic study can contribute to several promising directions, e.g., unsupervised/semi-supervised/weakly-supervised cross-mode learning. Source code and dataset are available at https://github.com/JingZhang617/cascaded_rgbd_sod.

3.1 Introduction

Saliency detection models are trained to discover the region(s) of an image that attract human attention. According to whether depth data is used, static image

[1]Code, results, and benchmarks will be made publicly available.

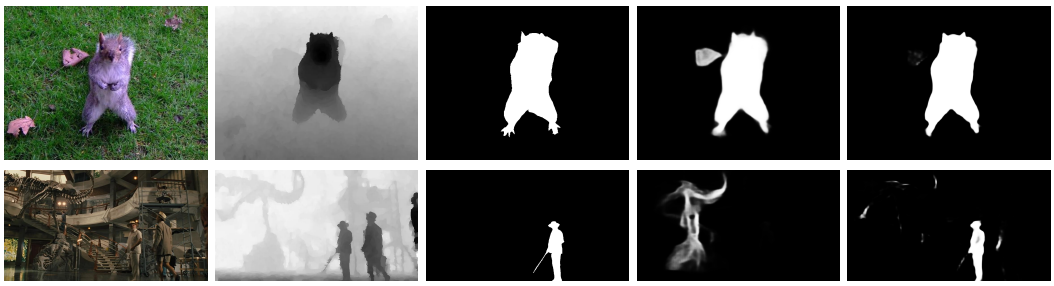


Figure 3.1: Comparison of saliency prediction of a state-of-the-art RGB-D saliency detection model, e.g., BBSNet [125], with ours.

Table 3.1: Comparison with the widely used datasets.

Dataset	Year	Size	Type	Depth Source	#Train	#Test
SSB [169]	2012	1,000	Internet	Stereo cameras+ optical flow [193]	-	1,000
NLPR [168]	2014	1,000	Indoor/Outdoor	Microsoft Kinect [194]	700	300
DES [171]	2014	135	Indoor	Microsoft Kinect [194]	-	135
NJU2K[167]	2014	1,985	Movie/Internet	FujiW3 camera + Sum's optical flow [195]	1,500	485
LFSD [170]	2014	80	Indoor/Outdoor	Lytro Illum cameras [196]	-	80
DUT [1]	2019	1,200	Indoor/Outdoor	Lytro2 camera+ [197]	800	400
SIP [43]	2020	929	Person in outside	Huawei Mate10	-	929
OUR	2021	**20,625**	Indoor/Outdoor	Holopix Social Platform [198]	**13,025**	**7,600**

saliency detection models can be divided into RGB image saliency detection models [33, 130, 34, 35, 134] and RGB-D saliency detection models [60, 1, 152, 131]. The former predicts saliency regions from an input RGB image, while the latter takes both the RGB image and depth data as input. With the availability of extra depth data as shown in Table 3.1, RGB-D saliency detection attracts great attention recently. Although huge performance improvement has been witnessed, none of the state-of-the-art (SOTA) methods model the procedure of complementary learning *explicitly*.

The RGB image provides appearance information, while the depth data introduces geometric information. Effective fusion of these two types of data can lead to a model that benefits from both modes. Towards this goal, existing RGB-D saliency detection models focus on fusing the information of these two modes. Three main fusion strategies have been widely studied for RGB-D saliency detection: early fusion [145, 60], late fusion [146, 49, 134] and cross-level fusion [1, 45, 147, 48, 131, 132, 133, 125, 150, 151, 152].

Although the three fusion strategies can learn from both RGB and depth data, there is no constraint in the network design to force a network to learn complementary information from the two modes. As a multi-mode learning task, a trained model should maximize the joint entropy of different modes within the network capacity. In other words, maximizing the joint entropy is also equal to the minimization of mutual information, which prevents a network from focusing on redundant

information. To explicitly model the complementary information between the RGB image and depth data, we introduce a latent-variable model based RGB-D saliency detection network with a mutual information minimization regularizer. Specifically, we design a "Complementary Learning" module as shown in Fig. 3.2 to achieve two main benefits: 1) explicitly modeling the redundancy between appearance features and geometric features; 2) fusing appearance features with depth features in latent space to achieve multi-mode fusion.

Furthermore, we observe that the existing RGB image saliency detection training datasets [41, 199] contain more than 10K images, while there is no large-scale RGB-D saliency detection training set. In Table 3.1 we compare the widely used RGB-D saliency datasets, in terms of the size, types of data, the sources of depth data, and their roles (for training "Tr" or for testing "Te") in RGB-D saliency detection. We note that the conventional training set for RGB-D saliency detection is a combination of samples from the NJU2K [167] dataset and NLPR [168], which includes only 2,200 image pairs in total. Although another 800 training images from the DUT dataset [1] can serve as the third part of the training set, the total number of training images is 3,000, which is quite small compared with existing RGB saliency detection training sets. In addition, we observe there are similar backgrounds in the existing RGB-D saliency training set, e.g., more than 10% of the training dataset comes from the same scene with similar illumination conditions. The lack of diversity in the dataset may render models with poor generalization ability. Moreover, the largest testing set [43] contains only 1,000 image pairs, which may not be enough to evaluate the overall performance of the deep RGB-D saliency detection models.

To provide a large RGB-D saliency detection dataset for robust model training, and a sufficient size of testing set for model evaluation, we contribute the largest-scale ($10\times$scale of previous) RGB-D saliency detection dataset, relabeled from Holo50K dataset [198], with 8,025 image pairs for training and 7,600 image pairs for testing. Importantly, we not only provide binary annotations, but also annotations for stereoscopic saliency detection, scribble and polygon annotations for weakly supervised RGB-D saliency detection, instance-level RGB-D saliency annotations and RGB-D saliency ranking. We also contribute 5,000 unlabeled training image pairs for semi-supervised or self-supervised RGB-D saliency detection. Our main contributions are:

- We design a latent variable model based RGB-D saliency detection network to explicitly model complementary information between the RGB image and depth data.

- We contribute the largest RGB-D saliency detection dataset (Ours), with a $\sim$15K labeled set and a 5K unlabeled set. For the labeled set, we provide five types of annotations for both fully-/weakly-/un-supervised RGB-D saliency detection.

- We present new benchmarks for RGB-D saliency detection, and introduce baseline models for stereoscopic and weakly supervised RGB-D saliency detection.

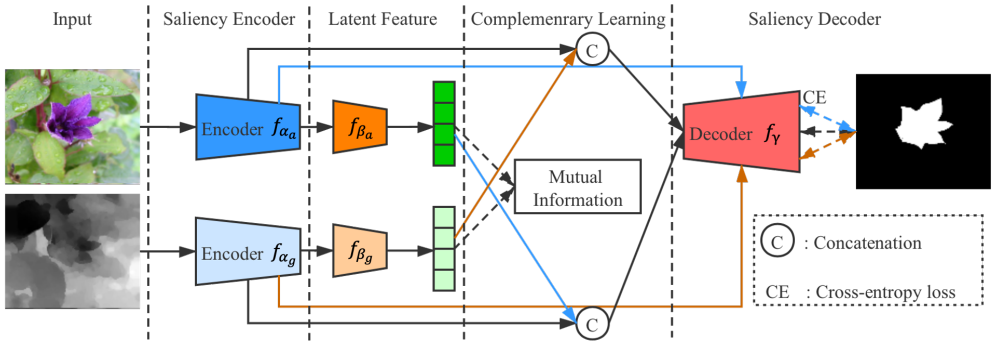


Figure 3.2: Overview of the proposed latent variable based complementary learning framework. Four main modules are included in our framework: 1) a "Saliency Encoder" module to extract feature from both the RGB image and the depth data; 2) a "Latent Feature" module to obtain latent features from each mode; 3) a "Complementary Learning" module to explicitly achieve complementary learning with the mutual information minimization constraint; and 4) a "Saliency Decoder" module to generate the predictions.

3.2 Related Work

3.2.1 RGB-D saliency datasets

The widely used RGB-D saliency detection datasets include NJU2K [167], NLPR [168], SSB [169], DES [171], LFSD [170], SIP [43], DUT [1], *etc.*, as shown in Table 3.1. The typical training dataset is the combination of 1,485 images from NJU2K [167] and 700 images from NLPR [168]. Piao *et al.* [1] introduces the DUT dataset, with 800 images for training and 400 images for testing. To prosper the RGB-D saliency detection task, we introduce the largest RGB-D saliency detection training and testing dataset, which will be introduced in Section 3.4.

3.2.2 RGB-D saliency models

For RGB-D saliency detection, one of the main focuses is to explore the complementary information between the RGB image and the depth data. The former provides appearance information of the scenario, while the latter introduces geometric information. Depending on how information from these two modes is fused, existing RGB-D saliency detection models can be divided into three categories: early-fusion models [145, 60], late-fusion models [146, 49, 134] and cross-level fusion models [1, 45, 147, 48, 131, 132, 133, 125, 150, 151, 152]. The first solution directly concatenates the RGB image with its depth information, forming a four-channel input, while the late fusion models treat each mode (RGB and depth) separately, and then fusion is achieved in the output layer. The above two solutions perform multi-mode fusion in the input or output, while the cross-level fusion models implement multi-mode fusion in the feature level. Specifically, features of an RGB image and depth

are gradually fused to effectively learn complementary information. Although those competing methods fuse the RGB image and depth data, none of them *explicitly* illustrate how the complementary information is learnt. We propose a cross-level fusion model as shown in Fig. 3.2. By designing the "**Complementary Learning**" module, we aim to reduce redundancy of appearance features and geometric features, and at the same time, we can effectively fuse features of the two modes in the latent space.

3.2.3 Latent variable models

Latent variable models [50, 52] are those with an extra latent variable embedded in the network to achieve stochastic predictions, which are widely used in image reconstruction and image synthesise [200, 201, 56], uncertainty estimation [202], *etc.*. In saliency detection, [60] uses a latent variable model to model the labeling variants for saliency distribution estimation. Li *et al.* [69] introduces superpixel-wise VAE [50] to reconstruct the background, and define the residual of the reconstructed background and the original image as salient foreground. A GAN [52] based framework is trained by [83] to achieve higher-order ground truth and a prediction similarity measure. The discriminator in [157] is designed to achieve multi-mode fusion. Here, we adopt the latent variable model to formulate the complementary information between the RGB image and depth data.

3.3 Proposed Method

We introduce a latent variable based complementary learning framework in Fig. 3.2 to explicitly model complementary information for RGB-D saliency detection.

3.3.1 Saliency encoder

Denote our training dataset as $T = \{X_i, Y_i\}_{i=1}^{N}$, where i indexes the images and N is the size of the training set, X_i and Y_i are the input RGB-D image pair and its corresponding ground-truth (GT) saliency map. We feed the training image pairs (RGB image I and the depth D) to the saliency encoder, as illustrated in Fig. 3.2, to extract appearance features $f_{\alpha_a}(I)$ and geometric features $f_{\alpha_g}(D)$ respectively, where α_a and α_g are the parameters of our RGB saliency encoder and depth saliency encoder respectively.

We build the saliency encoder upon the ResNet50 network [28], which includes four convolutional groups $\{s^1, s^2, s^3, s^4\}$. We add one additional convolutional layer of kernel size 3×3 after each $s^c \in \{s^c\}_{c=1}^{4}$ to reduce the channel dimension of s^c to $C = 32$, and obtain feature maps $\{e^1, e^2, e^3, e^4\}$. The final output of the saliency encoder module includes $e_a = \{e_a^1, e_a^2, e_a^3, e_a^4\}$ for the RGB image branch, and $e_g = \{e_g^1, e_g^2, e_g^3, e_g^4\}$ for the depth branch. Note that, the RGB saliency encoder and depth saliency encoder share the same network structure but not weights.

3.3.2 Latent feature

Given the output $e_a = \{e_a^1, e_a^2, e_a^3, e_a^4\}$ from the RGB saliency encoder and $e_g = \{e_g^1, e_g^2, e_g^3, e_g^4\}$ from the depth saliency encoder, the "Latent Feature" module is developed to map e_a and e_g to latent feature $z_a = f_{\beta_a}(e_a)$ and $z_g = f_{\beta_g}(e_g)$ respectively, where β_a and β_g are the parameters of the latent feature module for RGB images and depth respectively. Specifically, we first concatenate $\{e^c\}_{c=1}^4$ along channels to obtain a feature map of channel size $4 * C$, and then feed it to the latent feature net for latent feature extraction. The latent feature module is composed of five convolutional layers and two fully connected layers. The five convolutional layers have the same kernel size of 4×4 and stride size 2. The convolutional layers gradually map the appearance features or geometric features of channel size $4 * C$ to feature maps of channel size C, $2 * C$, $4 * C$, $8 * C$ and $2 * C$ respectively, and we obtain a $2 * C$ channel feature map in the end. Then two fully connected layers of size $K = 6$ are adopted to obtain the mean μ and standard deviation σ of the latent feature z following the variational auto-encoder (VAE) learning pipeline [50]. We reconstruct the latent feature z with reparameterization [50]: $z = \mu + \epsilon * \sigma$, where $\epsilon \sim N(0, 1)$.

3.3.3 Complementary learning

After obtaining the latent features z_a and z_g for the RGB image and depth data, we introduce a mutual information minimization regularizer to explicitly reduce the redundancy between these two modes. Our basic assumption is that a good appearance saliency feature and geometric saliency feature pair should carry both common parts (semantic related) and different attributes (domain related). Mutual information M_I is used to measure the difference between the entropy terms:

$$M_I(z_a, z_g) = H(z_a) + H(z_g) - H(z_a, z_g), \tag{3.1}$$

where $H(.)$ is the entropy, $H(z_a)$ and $H(z_g)$ are marginal entropies, and $H(z_a, z_g)$ is the joint entropy of z_a and z_g. Intuitively, we have the Kullback–Leibler divergence (KL) of the two latent variable (or the conditional entropies) as:

$$KL(z_a||z_g) = H(z_a, z_g) - H(z_a), \tag{3.2}$$

$$KL(z_g||z_a) = H(z_a, z_g) - H(z_g). \tag{3.3}$$

Combing Eq. 3.1, Eq. 3.2 and Eq. 3.3 we obtain:

$$M_I(z_a, z_g) = H(z_a, z_g) - (KL(z_a||z_g) + KL(z_g||z_a)). \tag{3.4}$$

Given the RGB image and the depth data, $H(z_a, z_g)$ is non-negative and fixed, then minimizing the mutual information can be achieved by minimizing the negative symmetric KL term: $\mathcal{L}_{mi} = -(KL(z_a||z_g) + KL(z_g||z_a))$. Intuitively, $M_I(z_a, z_g)$ is the reduction of uncertainty in z_a when z_g is observed, or vice versa. As a multi-

mode learning task, each mode should learn some new attributes of the task from other modes. Thus, by minimizing $M_I(z_a, z_g)$, we can effectively explore the complementary attributes of both modes.

Moreover, as z_a encodes the appearance information, and z_g encodes the geometric information, we intend to fuse the appearance feature and geometric feature in the latent space to achieve effective multi-mode fusion. Specifically, we map e_a^4 from the RGB saliency encoder branch to a $K = 32$ dimensional feature vector by using one fully connected layer. Then we concatenate it with z_g, and map the concatenated feature with one DenseASPP [162] to obtain the RGB saliency prediction P_a. Similarly, we can obtain the depth saliency prediction P_g by fusing e_g^4 with z_a.

3.3.4 Saliency decoder

With the complementary learning branch, we obtain RGB saliency prediction P_a and depth saliency prediction P_g with latent features from depth and the RGB image respectively. The saliency decoder f_γ takes the saliency features from the saliency encoder branches in Fig. 3.2, as well as P_a and P_g as input to compute our final prediction, where γ is the parameter set of the saliency decoder. Specifically, with the output $e_a = \{e_a^1, e_a^2, e_a^3, e_a^4\}$ from the RGB saliency encoder and $e_g = \{e_g^1, e_g^2, e_g^3, e_g^4\}$ from the depth saliency encoder, we add a position attention module and a channel attention module [144] after each $\{e_a^c\}_{c=1}^4$ and $\{e_g^c\}_{c=1}^4$. Then we concatenate the four groups of feature maps after the dual attention and feed it to the DenseASPP [162] to obtain our saliency prediction P_f. To further fuse information from both modes, we concatenate P_a, P_g and P_f channel-wise, and feed it to a 3×3 convolutional layer to achieve our final prediction P.

3.3.5 Objective function

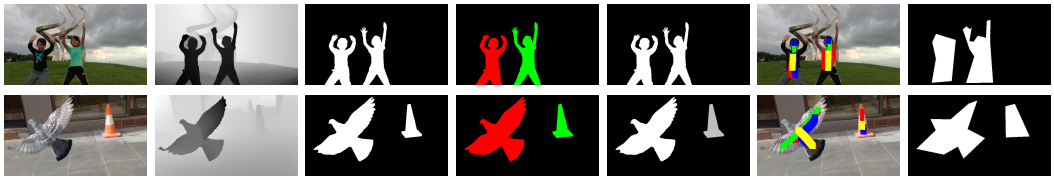
We adopt the binary cross-entropy loss $\mathcal{L}_{ce}$ as our objective function to train our latent variable model based complementary learning framework, where the complementary constraint, as indicated in Eq. 3.1, pushes the saliency feature distribution of the RGB image to be apart from that of the depth data. Our final objective function is:

$$\mathcal{L} = \mathcal{L}_{ce}(P, Y) + \lambda_1 \mathcal{L}_{ce}(P_f, Y) + \lambda_2 \mathcal{L}_{ce}(P_a, Y) + \lambda_3 \mathcal{L}_{ce}(P_g, Y) + \lambda \mathcal{L}_{mi}(z_a, z_g), \quad (3.5)$$

and empirically we set $\lambda_1 = 0.8, \lambda_2 = 0.6, \lambda_3 = 0.4, \lambda = 0.1$.

3.4 Newly Collected Dataset

Existing training images for RGB-D saliency detection (the standard composite training dataset) come from two main datasets: (1) NJU2K [167] and (2) NLPR [168]. Piao *et al.* [1] introduces an additional 800 images for training and another 400 images for testing to the DUT dataset. As suggested by [1], to test model performance



(a) (b) (c) (d) (e) (f) (g)

Figure 3.3: Annotations of our new RGB-D saliency detection datasets: (a) the RGB image, (b) the depth data and (c) the binary ground truth, (d) the instance level annotation, (e) the ranking based annotation, (f) the scribble annotation and (g) the polygon annotation. Our diverse annotations will facilitate developing different fully/weakly supervised RGB-D saliency detection.

on the DUT testing dataset, one needs to train with the combination of the NJU2K and NLPR training sets, and fine-tune the model on the DUT training dataset. We argue that the limited size of the training set compared with RGB saliency detection[2] may lead to models with poor generalization ability. Furthermore, different splits of a training set often lead to inconsistent performance evaluation. To further boost RGB-D saliency detection, we contribute the largest RGB-D saliency detection dataset. Moreover, we provide binary annotation, instance level annotation, ranking based annotation, weak annotation as shown in Fig. 3.3.

3.4.1 Dataset annotation

Our new Ours is based on Holo50K [198], which is a stereo dataset. We select 15,625 stereo image pairs from it to be labeled (the candidate labeled set) and another 5,000 image pairs as the unlabeled set. Note that the stereo pairs in Holo50K dataset are directly captured by a stereo camera without rectification, so we use a SOTA off-the-shelf optical flow algorithm [203] to compute the pseudo depth of both the candidate labeled set and unlabeled set with the left-right view images as input.

To provide annotations for the candidate labeled set, we firstly ask five "coarse" annotators to coarsely label each image (only the right view image is used) with scribble annotations according to their own preference of saliency. Secondly, the "fine" annotators will segment the full scopes of salient objects and provide instance-level annotations. Thirdly, we perform majority voting to obtain the binary GT saliency maps for our RGB-D saliency detection task. Moreover, based on the scribble annotations and instance-level saliency maps, we rank each saliency instance according to the initial scribble annotations to form our RGB-D saliency ranking dataset.

We also provide weak annotations for weakly-supervised RGB-D saliency detection, including scribble annotations and polygon annotations as shown in Fig. 3.3. We define majority of scribble annotations from multiple coarse annotators as the scribble annotations of our dataset. Specifically, we first obtain the instance with the

[2]The two largest RGB saliency training datasets have 10,553 (DUTS [41]) and 10,000 (MSRA10K [199]) images respectively.

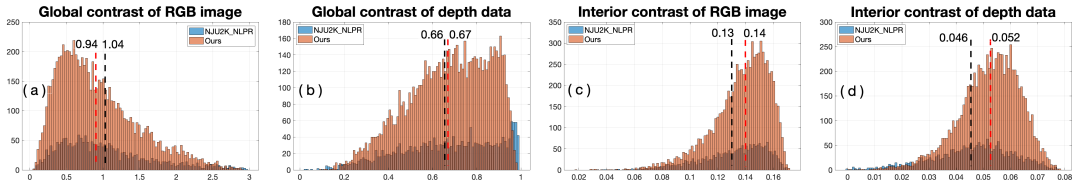


Figure 3.4: Global contrast of RGB images (a) and depth images (b). Interior contrast of RGB images (c) and depth images (d).

majority of scribble. Then, we define the scribble on the majority instance as our scribble annotation. We label the majority salient instance with polygons to form our polygon based annotations.

3.4.2 Dataset analysis

Contrast analysis: As a contrast based task, an effective training dataset for RGB-D saliency detection should include a wide range of images with salient objects of different contrast. Hence, we compute the global and interior contrast of both the RGB images and depth images of our new training dataset and the existing training dataset as shown in Fig. 3.4(a)-(d), where the x-axis is the corresponding contrast and the y-axis represents the number of images. Global contrast measures the saliency of the object, indicating the noticeability of the salient object, while the interior contrast measures the consistency inside the same salient object, representing the intra-class consistency. To obtain global contrast, we compute the H dimensional color histogram[3] of both the salient foreground and background. Then we adopt Chi-squared distance to measure the global contrast between the salient object and background. We define the mean of the Chi-squared distance as the image global contrast. For interior contrast, we compute the entropy of the color histogram of the salient object.

As shown in Fig. 3.4 (a)-(b), we obtain smaller global contrast for RGB images and higher global contrast for depth maps. The small RGB global contrast indicates the greater difficult of the new training images, while the high depth global contrast indicates that our depth is a good option complementary to the RGB images for salient object detection. Further, the higher interior contrast of both our training RGB images and depth in Fig. 3.4 (c) and (d) indicates the new dataset is challenging.

Depth quality estimation: The quality of the depth plays an important role for RGB-D saliency detection. We adopt two evaluation metrics to measure the depth quality. Firstly, we use the smoothness error [109] to measure the edge alignment of the depth. Then we use the warping error [203] to measure the correctness of depth. The former highlights the structure of depth, while the latter focuses on the overall accuracy of depth.

The smoothness error was originally used for occlusion-aware flow estimation

[3]Following [204], we set $H = 16$ dimensional histogram for the Red, Green and Blue channel of the RGB image respectively, and the color histogram is then the concatenation of above histograms.

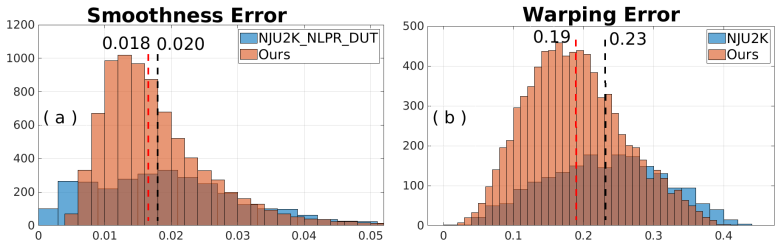


Figure 3.5: (a) Smoothness error and (b) Warping error of the new dataset compared with the standard composite training dataset.

[109]. Given input image I and depth D, we define the smoothness error as:

$$\mathcal{L}_s = \sum_{u,v} \sum_{d \in \overrightarrow{x}, \overrightarrow{y}} \Psi(|\partial_d D_{u,v}| e^{-\alpha |\partial_d I(u,v)|}), \tag{3.6}$$

where Ψ is defined as $\Psi(s) = \sqrt{s^2 + 1e^{-6}}$, $D_{u,v}$ is the depth at position (u, v), and $I(u, v)$ is the image intensity, d indexes over partial derivatives in the $\overrightarrow{x}$ and $\overrightarrow{y}$ directions. We set $\alpha = 10$ following the setting in [109].

For each image, we compute its mean smoothness error, and show the distribution of the smoothness errors for our training set and the combination of existing training sets (NJU2K [167], NLPR [168], and DUT [1]) in Fig. 3.5 (a). The relatively smaller mean smoothness error indicates better edge-alignment of our depth maps.

We further adopt the photometric warping loss as another metric to evaluate the quality of depth from the perspective of warping errors between the stereo image pair. As the warping error measures reconstruction errors from one view to the other one based on the given depth, we only compare with NJU2K [167], as it is the only training set that contains both left and right view images. We manifest the warping errors of our Ours and NJU2K in Fig. 3.5 (b). The smaller warping error of our training set further indicates better depth quality of our dataset.

3.4.3 High-quality Diverse Annotation

A high-quality dataset should have samples from diverse scenes and objects of diverse categories. Further, the annotation should be precise. We then analysis the diversity of the dataset, and show more of our diverse annotations.

To analyse the diversity of the scene, we feed images of our entire dataset to an existing scene classification network [205] to predict the scene category for each image. Further, with our instance level annotation, we obtain the instance pool of our dataset, which is then fed to an existing image classification network to predict object level category. Note that, as there exists no "Person" category in ImageNet, we test with image classification model trained on Microsoft COCO dataset to predict category of each instance in our dataset. We show the object category distribution and scene category distribution in Fig. 3.6, which clearly shows that our dataset contains

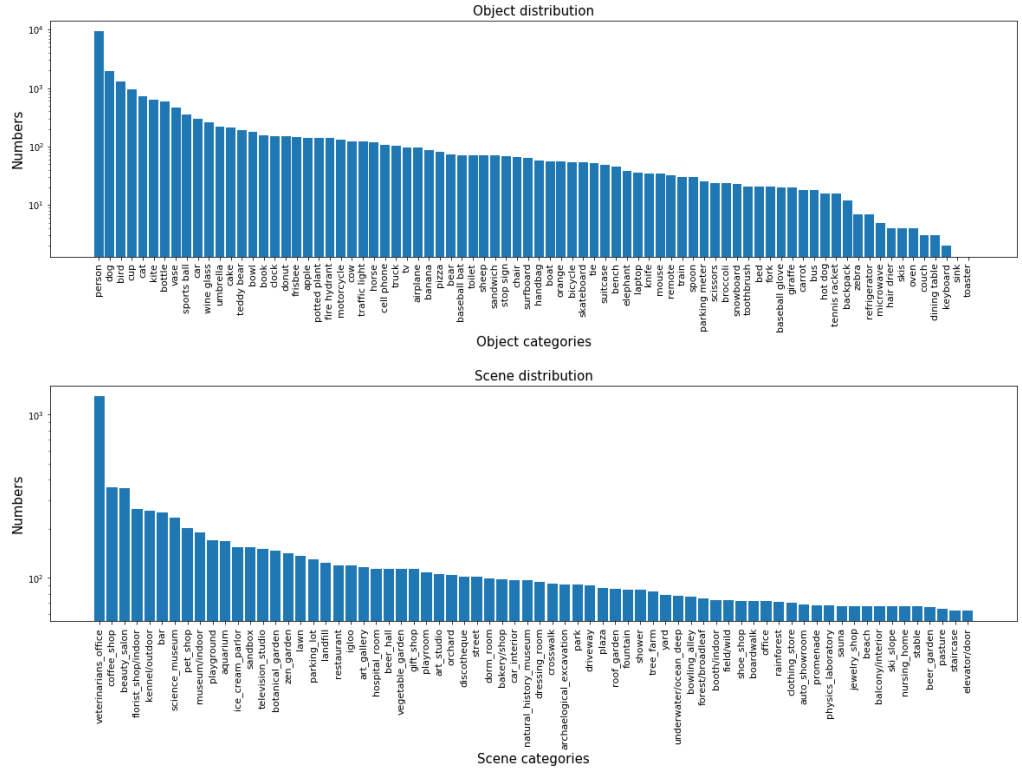


Figure 3.6: Object distribution (top) and scene distribution (bottom) of our new dataset.

various scenes and diverse objects. In addition, for the scene classification model, it produces the indoor/outdoor category distribution as well, which indicates a 57% indoor scenes and 43% outdoor scenes. The balanced indoor/outdoor distribution further illustrates the high-quality of our new dataset.

Our dataset is labeled following a four-step process. Firstly, the coarse annotators label the whole dataset with scribble annotation. Then, the fine annotators segment the full scope of each instance with scribble on it. Thirdly, we check the annotation. Lastly, the low-quality annotations will be sent back to the fine annotators to label it again. In Fig. 3.7 we show five samples of low-quality which are re-labeled ("Rejected"), and high-quality after the third step ("Accepted"). Our main criteria is that we want to keep all the instances with scribble on them to be labeled. Further, we want that all the instances are distinguishable, which will makes our dataset effective for instance saliency detection.

3.4.4 More Statistics

We provide the complete information of the existing RGB-D saliency datasets and the proposed dataset in Table 3.1. It clearly shows that our dataset is the largest-scale dataset with more diverse annotations.

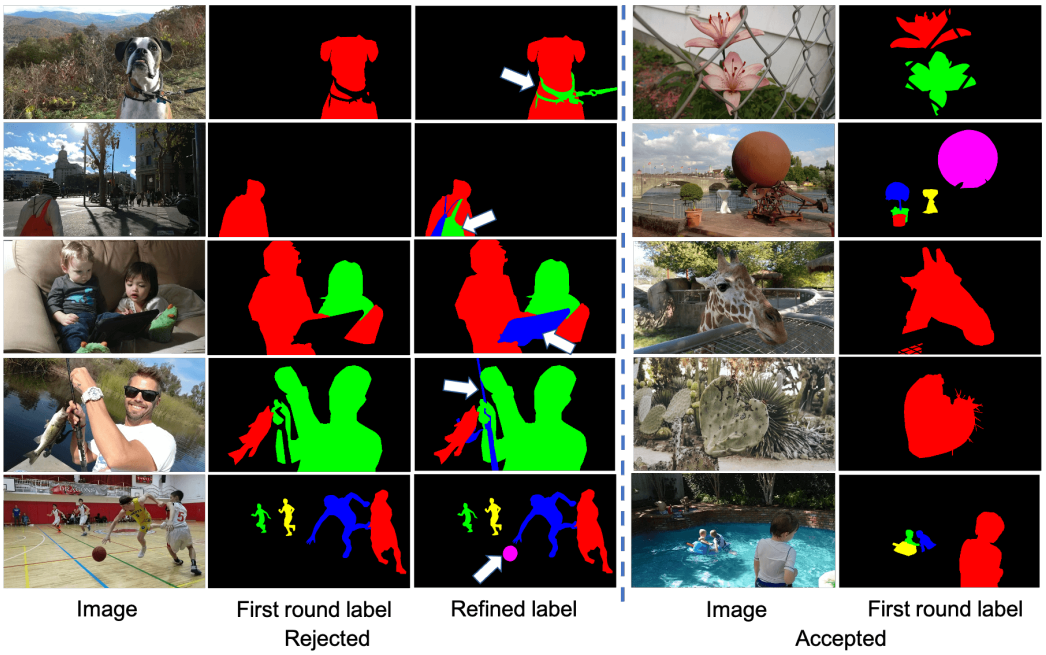
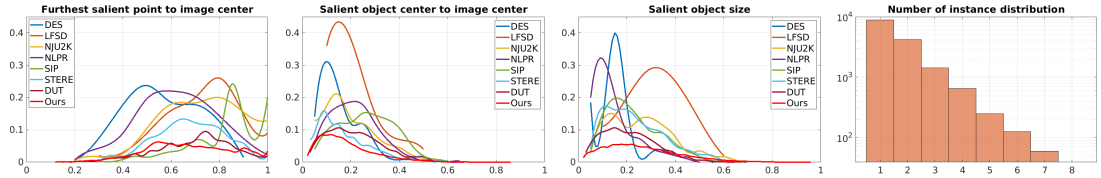


Figure 3.7: Visualization of our annotation. The "Rejected" samples are relabeled to obtain high-quality annotation as the "Accepted" ones.



(a) (b) (c) (d)

Figure 3.8: More statistics of our new dataset. (a) Distribution of the furthest salient points to image center. (b) Distribution of salient objects to image center. (c) Distribution of salient object sizes. (d) Number of salient instances.

Further, we analyse the center bias of our dataset, the size of the salient objects, the number of salient objects, and show the results in Fig. 3.8 (a)-(d). Center bias is a common artifact of sense for salient objects, which indicates that salient objects are usually located in the center of the image. We follow [191] and use the "furthest salient point to image center" and "salient object center to image center" as metrics to evaluate the center-bias of existing datasets and ours. For both the above center based curves, the x-axis is the distance, and y-axis is the probability. According to [191], the smaller probability of both center based metrics indicate the less center bias, and Fig. 3.8 (a)-(b) clearly show that our new dataset suffers less from the center bias. The size of the salient object is defined as the proportion of salient pixels in the image. Fig. 3.8 (c) shows that salient objects in our new dataset varies in a broader

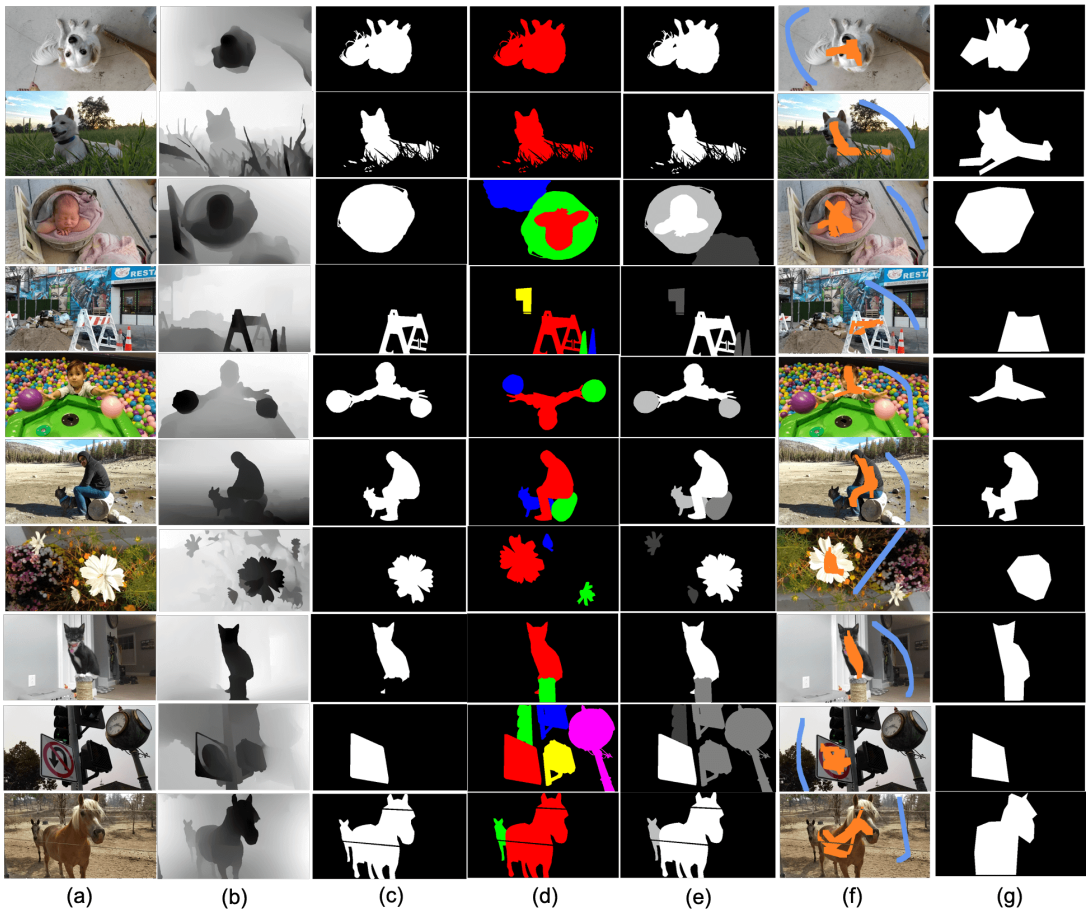


Figure 3.9: Annotations of our new RGB-D saliency detection datasets. From left to right: (a) the RGB image, (b) the depth data, (c) the binary ground truth, (d) the instance level annotation, (e) the ranking based annotation, (f) the scribble annotation and (g) the polygon annotation. Our diverse annotations will facilitate developing different fully/weakly supervised RGB-D saliency detection.

range. In Fig. 3.8 (d), we show the number of instances in each image of our new dataset. The wide distribution of instance number indicates the complexity of our new dataset.

3.4.5 Dataset Visualization

We provide five different types of annotations, including binary ground truth, instance level annotation, ranking based annotation, scribble annotation and polygon annotation. We show more samples from our dataset in Fig. 3.9. Compared with conventional binary ground truth, the extra annotations can further boost related tasks for both fully and weakly supervised saliency detection.

3.4.6 Dataset Splitting

Our new dataset has 20,625 samples, where 15,625 samples are labeled, and the other 5,000 samples are unlabeled. We then divide the labeled set into one training set with 8,025 samples and two different testing sets of size 4,600 and 3,000 respectively, namely the "Normal" one and the "Difficult" one. The training dataset is generated by randomly select 8,025 images from the labeled set. For the testing datasets, we intend to introduce two sets of different difficulty. Specifically, we rank the RGB images based on both global and interior contrast, and define samples of low global contrast and high interior contrast be the difficult samples. Then we have a pool of difficult samples D_d and normal samples D_n, with size of 1,800 and 5,800 respectively. We random select 30% samples from D_d and 70% samples from D_n to obtain our "Normal" testing set, and the other samples form our "Difficult" set.

3.5 Experiments

We compare the proposed complementary learning framework with competing RGB-D saliency detection models, and report the performance in Table 3.2 & 3.3. Furthermore, we retrain the state-of-the-art RGB-D saliency detection models on our new training dataset, and provide the performance of those models on our testing dataset in Table 3.4. We also explore our dataset by providing three benchmark and baseline models on our weak annotations and stereoscopic saliency dataset.

3.5.1 RGB-D saliency detection

Dataset: For fair comparisons with existing RGB-D saliency detection models, we follow the conventional training setting, in which the training set is a combination of 1,485 images from the NJN2K dataset [167] and 700 images from the NLPR dataset [168]. We then test the performance of our model and competing models on the NJU2K testing set, NLPR, testing set LFSD [170], DES [171], SSB [169] SIP [43] and DUT [1] testing set.

Metrics: We evaluate the performance of the models on four golden evaluation metrics, i.e., , Mean Absolute Error ($\mathcal{M}$), Mean F-measure (F_β), Mean E-measure (E_ξ) [173] and S-measure (S_α) [172].

Training details: Our model is trained in Pytorch using the ResNet50 [28] as backbone as shown in Fig. 3.2. The encoders of RGB and depth share the same network structure, and are initialized with ResNet50 [28] trained on ImageNet, and other newly added layers are randomly initialized. We resize all the images and ground truth to the same spatial size of 352×352 pixels. We set the maximum epoch as 100, and initial learning rate as 5e-5. We adopt the "step" learning rate decay policy, and set the decay size as 80, and decay rate as 0.1. The whole training takes 4.5 hours with batch size 5 on an NVIDIA GeForce RTX 2080 GPU.

Quantitative comparison: We compare the performance of our model and state-of-the-art RGB-D saliency detection models, and report the performance in Table 3.2

Table 3.2: Benchmarking results of leading handcrafted feature-based models and deep models (*) on six RGB-D saliency datasets. ↑ & ↓ denote the larger and smaller is better, respectively. Here, we adopt mean F_β and mean E_ξ[173].

		Early Fusion Models			Late Fusion Models				Cross-level Fusion Models								
	Metric	DANet [206]*	UCNet [60]*	JLDCF [47]*	CDB [184]	A2dele [134]*	AFNet [146]*	CTMF [49]*	DMRA [1]*	TANet [48]*	CPFP [131]*	S2MA [133]*	BBS-Net [125]*	CoNet [150]*	HDFNet [151]*	BiaNet [152]*	Ours Ours*
NJU2K	$S_\alpha \uparrow$	.897	.897	.902	.632	.873	.822	.849	.886	.879	.878	.894	.921	.911	.908	.915	**.933**
	$F_\beta \uparrow$	.877	.886	.885	.498	.867	.827	.779	.873	.841	.850	.865	.902	.903	.892	.903	**.916**
	$E_\xi \uparrow$	.926	.930	.935	.572	.913	.867	.846	.920	.895	.910	.914	.938	.944	.936	.934	**.949**
	$\mathcal{M} \downarrow$	.046	.043	.041	.199	.051	.077	.085	.051	.061	.053	.053	.035	.036	.038	.039	**.034**
SSB	$S_\alpha \uparrow$	.892	.903	.903	.615	.876	.825	.848	.835	.871	.879	.890	.908	.896	.900	.904	**.915**
	$F_\beta \uparrow$	.857	.884	.873	.489	.874	.806	.758	.837	.828	.841	.853	.883	.877	.870	.879	**.887**
	$E_\xi \uparrow$	.915	.938	.936	.561	.925	.872	.841	.879	.893	.911	.914	.928	.939	.931	.926	**.943**
	$\mathcal{M} \downarrow$	.048	.039	.040	.166	.044	.075	.086	.066	.060	.051	.051	.041	.040	.041	.043	**.036**
DES	$S_\alpha \uparrow$	.905	.934	.931	.645	.881	.770	.863	.900	.858	.872	.941	.933	.906	.926	.931	**.947**
	$F_\beta \uparrow$	.848	.919	.907	.502	.868	.713	.756	.873	.790	.824	.909	.910	.880	.910	.910	**.928**
	$E_\xi \uparrow$	.961	.967	.959	.572	.913	.809	.826	.933	.863	.888	.952	.949	.939	.957	.948	**.973**
	$\mathcal{M} \downarrow$	.028	.019	.021	.100	.030	.068	.055	.030	.046	.038	.021	.021	.026	.021	.021	**.016**
NLPR	$S_\alpha \uparrow$	.908	.920	.925	.632	.887	.799	.860	.899	.886	.888	.916	.930	.900	.923	.925	**.935**
	$F_\beta \uparrow$	.850	.891	.894	.421	.871	.755	.740	.865	.819	.840	.873	.896	.859	.894	.894	**.902**
	$E_\xi \uparrow$	.945	.951	.955	.567	.933	.851	.840	.940	.902	.918	.937	.950	.937	.955	.948	**.958**
	$\mathcal{M} \downarrow$	.031	.025	.022	.108	.031	.058	.056	.031	.041	.036	.030	.023	.030	.023	.024	**.020**
LFSD	$S_\alpha \uparrow$	.845	.864	.862	.520	.831	.738	.796	.847	.801	.828	.837	.864	.842	.854	.845	**.867**
	$F_\beta \uparrow$	.826	.855	.848	.376	.829	.736	.756	.845	.771	.811	.806	.843	.834	.835	.834	**.856**
	$E_\xi \uparrow$	.872	.901	.894	.465	.872	.796	.810	.893	.821	.863	.855	.883	.886	.883	.871	**.903**
	$\mathcal{M} \downarrow$	.082	.066	.070	.218	.076	.134	.119	.075	.111	.088	.094	.072	.077	.077	.085	**.064**
SIP	$S_\alpha \uparrow$	.878	.875	.880	.557	.826	.720	.716	.806	.835	.850	.872	.879	.868	.886	.883	**.889**
	$F_\beta \uparrow$	.829	.867	.873	.341	.827	.702	.608	.811	.803	.821	.854	.868	.855	.875	.873	**.882**
	$E_\xi \uparrow$	.914	.914	.918	.455	.887	.793	.704	.844	.870	.893	.905	.906	.915	.923	.913	**.928**
	$\mathcal{M} \downarrow$	.054	.051	.049	.192	.070	.118	.139	.085	.075	.064	.057	.055	.054	.047	.052	**.046**

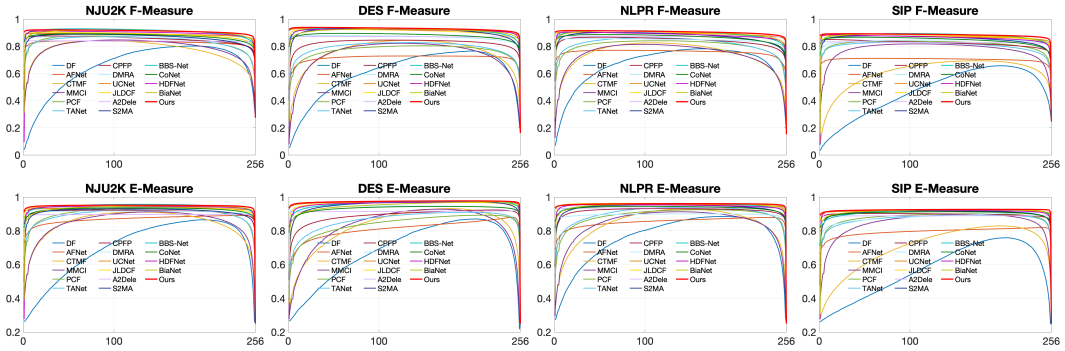


Figure 3.10: F-measure and E-measure curves on four testing datasets (first row: F-measure curves; second row: E-measure curves).

and Fig. 3.10. Note that, we use the training set of NJU2K and NLPR as competing deep RGB-D saliency detection models. We observe that performance differences of current RGB-D saliency detection are very subtle, e.g., HDFNet [151], BiaNet [152], and CoNet [150]. The consistently better performance of our model indicate the effectiveness of our solution.

Table 3.3: Model performance on DUT [1] testing set.

Metric	UCNet [60]	JLDCF [47]	A2dele [134]	DMRA [1]	CPFP [131]	S2MA [133]	CoNet [150]	HDFNet [151]	Ours Ours
$S_\alpha \uparrow$	.907	.905	.884	.886	.749	.903	.919	.905	**.925**
$F_\beta \uparrow$	.902	.884	.889	.883	.695	.881	.911	.889	**.918**
$E_\xi \uparrow$	.931	.932	.924	.924	.759	.926	.947	.929	**.956**
$\mathcal{M} \downarrow$	.038	.043	.043	.048	.100	.044	.033	.040	**.031**

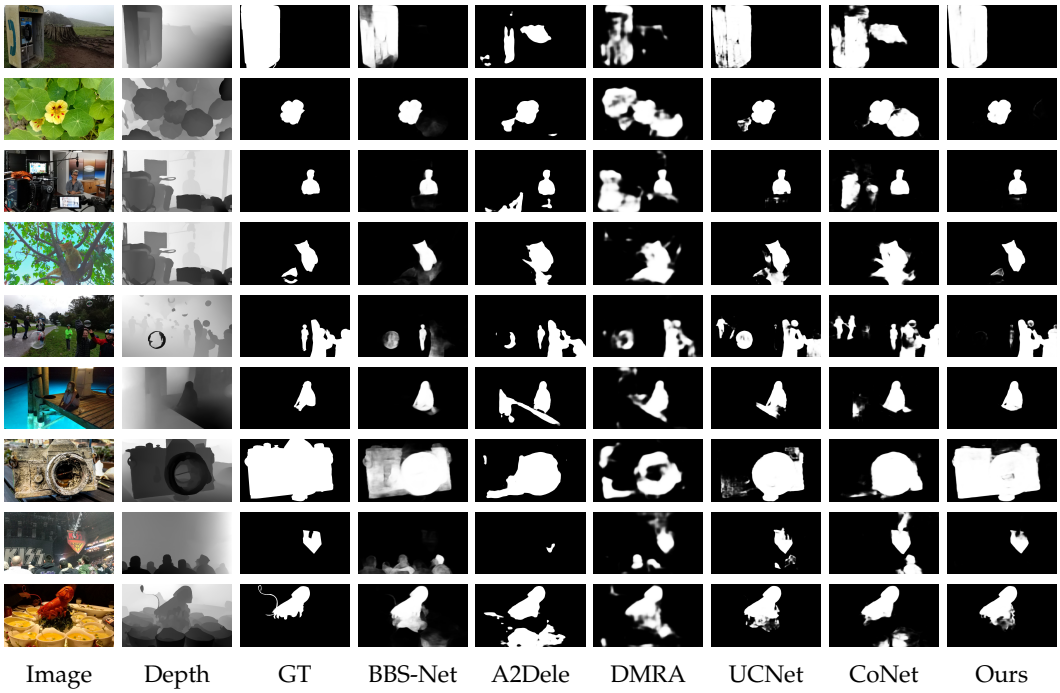


Figure 3.11: Performance comparison with state-of-the-art methods on our new testing dataset.

Performance on the DUT [1] dataset: Some existing RGB-D saliency detection approaches [1, 133] fine-tune their models on the DUT training dataset [1] to evaluate their performance on the DUT testing set. To test our model on the DUT testing set, we follow the same training strategy. In Table 3.3, all the models are trained with the conventional training set and then fine-tuned on the DUT training set. The consistently superior performance further illustrates the superiority of our model. Furthermore, since the current testing performance is achieved in a train-retrain manner, we re-train these models with a combination of the conventional training set and DUT as the training set, and observe consistently worse performance in this case. This observation tells us that inconsistent annotations may occur in these three training sets (i.e., NJU2K, NLPR and DUT). It also motivates us to collect a larger training dataset (Ours) with consistent annotations for robust model training. In Table 3.4, we retrain existing RGB-D saliency detection models with out new training dataset, and show their performance on our new testing datasets. The better performance

Table 3.4: Performance on our new testing datasets.

	Metric	UCNet [60]	JLDCF [47]	A2dele [134]	DMRA [1]	CPFP [131]	S2MA [133]	CoNet [150]	BBS-Net [125]	Ours Ours
Normal	$S_\alpha \uparrow$	.894	.894	.833	.782	.795	.877	.820	.902	**.906**
	$F_\beta \uparrow$	**.883**	.875	.835	.744	.716	.829	.796	.879	**.883**
	$E_\xi \uparrow$	**.929**	.919	.882	.812	.801	.881	.850	.923	.924
	$\mathcal{M} \downarrow$	**.036**	.042	.060	.105	.104	.059	.082	.039	**.036**
Difficult	$S_\alpha \uparrow$	.822	.845	.787	.743	.770	.828	.779	.853	**.859**
	$F_\beta \uparrow$	.814	.832	.795	.724	.704	.789	.774	.834	**.843**
	$E_\xi \uparrow$	.859	.870	.838	.775	.776	.836	.813	.876	**.887**
	$\mathcal{M} \downarrow$	.079	.075	.092	.137	.131	.092	.113	.071	**.068**

Table 3.5: Performance of the ablation study models.

	NJU2K[167]				SSB [169]				DES [171]				NLPR [168]				LFSD [170]				SIP [43]			
Method	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$
K3	.928	.908	.947	**.032**	.909	.892	.939	**.036**	.934	.922	.964	.018	.925	.904	.956	.022	**.869**	.845	.898	.067	.885	.879	.919	.047
K32	.924	.909	.944	.033	.908	**.894**	.941	**.036**	.938	.923	.966	.017	.927	**.906**	**.959**	.021	.856	.853	.900	.065	.885	.878	.921	.046
SS	.916	.907	.943	.034	.899	.882	.932	.040	.936	.927	.968	.017	.920	.896	.954	.024	.861	.852	.889	.077	.885	.876	.920	.047
W0	.918	.907	.944	.033	.892	.877	.923	.042	.934	.924	.964	.017	.924	.900	.945	.023	.843	.836	.881	.076	.884	.878	.916	.048
W1	.919	.909	.946	**.032**	.905	.886	.937	.037	.938	.927	.971	**.016**	.923	.903	.956	.022	.857	.853	.891	.071	.887	**.882**	.921	**.045**
Ours	**.933**	**.916**	**.949**	.034	**.915**	.887	**.943**	**.036**	**.947**	**.928**	**.973**	**.016**	**.935**	.902	.958	**.020**	.867	**.856**	**.903**	**.064**	**.889**	**.882**	**.928**	.046

of our solution further illustrate effectiveness of our solution. Further, the huge performance gap between those existing techniques further show necessary of our new training dataset.

Qualitative comparison: We further visualize our prediction and other in Fig. 3.1. The qualitative comparisons demonstrate that with the complementary learning strategy, our model can explore better complementary information for effective multi-mode learning. More results are shown in the supplementary materials.

3.5.2 Ablation study

Three main factors may influence the performance of our model, including: (1) the dimension of the latent space; (2) the structure of the "Latent Feature" module; and (3) the weight of the mutual information regularizer term in Eq. 3.5. We then perform three main ablation studies to further analyse the components of our model.

The dimension of the latent space: We set the dimension of latent space as $K = 6$. To test the impacts of different dimensions of the latent space on the network performance, we set $K = 3$ and $K = 32$, and report their performance as "K3" and "K32" respectively in Table 3.5. The experimental results demonstrates that our model achieves relative stable performance with different dimensions of the latent space. This is because the features from the "Saliency Encoder" module are representative.

The structure of the "Latent Feature" module: As discussed in Section 3.3.2, the "Latent Feature" module is composed of five convolutional layers and two fully connected layers for latent feature extraction. One may also achieve latent feature extraction directly from the output of the "Saliency Encoder". Specifically, we can use two

Table 3.6: Performance of the stereo saliency detection baseline.

NJU2K[167]			NJU400[167]			Ours-Normal			Ours-Difficult		
$S_\alpha \uparrow$	$F_\beta \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$\mathcal{M} \downarrow$
.874	.851	.056	.882	.851	.044	.874	.855	.047	.825	.812	.080

fully connected layers to map the concatenation of $\{e^c\}_{c=1}^4$ to μ and σ. In Table 3.5, we report the performance of our model with this simple setting, marked as "SS". We observe the performance decreases, which indicates necessity of introducing more non-linearity to effectively extract the latent feature of each mode.

The weight of the mutual information regularizer: The weight λ of the mutual information regularization term controls the level of complementary information. Empirically, we set $\lambda = 0.1$. We then test how the model performs with smaller or larger λ, and set $\lambda = 0$ and $\lambda = 1$ respectively. We show the performance of those variants in Table 3.5, denoted by "W0" and "W1". The inferior performance of "W0" indicates the effectiveness of our complementary information modeling strategy. Further, compared with our model, we observe relatively worse performance of "W1", which indicates the mutual information regularizer can indeed influence model performance. We will investigate a better strategy to adaptively set the weight of the mutual information in the future.

3.5.3 New Benchmarks on our dataset

The straightforward way to use our training dataset is using it for RGB-D saliency detection. Due to the multiple annotations provided in our new dataset as shown in Fig. 3.4, we further discuss three benchmarks that would need our annotations. We believe that our rich labels can motivate future model design.

Benchmark #1: Stereo saliency detection. As our RGB-D saliency dataset is constructed on a stereo dataset [198], we directly train a stereo image pair based saliency object detection model, where the depth is implicitly instead of explicitly obtained from the stereo image pairs. Although there exist some stereoscopic saliency detection models [207, 177, 169, 208], all of them take both the image(s) and depth as input. The video fixation prediction work in [209] introduces an intrinsic depth saliency estimation model for video fixation prediction without explicitly obtaining the depth data. Similar to [209], we design a real[4] stereoscopic saliency detection model, and provide a baseline to manifest the potential of our dataset for stereoscopic saliency detection. There are two existing stereoscopic saliency detection datasets, i.e., , the NJU2K testing set [167] and NJU400 [167] dataset, which include 500 and 400 left-right view image pairs respectively. Together with our new testing sets, we have four stereoscopic saliency detection testing sets as shown in Table 3.6. We train our stereoscopic saliency detection model on our new training dataset, and the left-right view images are taken as inputs and the GT saliency maps in the right view images

[4]We define the stereoscopic saliency models taking input only the left and right view images as the "real" stereoscopic saliency models.

Table 3.7: Performance of the weakly supervised saliency detection baselines.

	NJU2K[167]				SSB[169]				NLPR [168]				SIP [43]				Ours-Normal				Ours-Difficult			
Method	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$
Scribble	.823	.806	.869	.080	.820	.803	.884	.073	.820	.737	.863	.058	.815	.793	.888	.076	.802	.780	.856	.082	.767	.749	.812	.115
Polygon	.847	.827	.896	.065	.853	.831	.913	.056	.848	.789	.899	.043	.846	.822	.909	.060	.827	.805	.884	.065	.786	.774	.841	.096

are used as supervision. We then adopt the same "Saliency Encoder" and "Saliency Decoder" as in Fig. 3.2. Instead of explicitly using the depth data, we implicitly model the geometric information by using a cost volume between the saliency encoder and decoder. We then show its performance in Table 3.6, and we will explain the architecture in details in the supplementary material.

Weakly-supervised RGB-D saliency detection. Inspired by [63], we also introduce scribble and polygon (see Fig. 3.3 (f) and (g)) based weakly supervised RGB-D saliency detection networks to further explore our new dataset for weakly supervised learning.

Benchmark #2: Scribble based supervision. As no structure information exists in scribble annotations, we follow [63] and use the smoothness loss and an auxiliary edge detection branch as a constraint to maintain structure information in the prediction. Specifically, as our initial scribble annotations locate inside the salient objects, we then create extra background scribble annotations following [63]. In this case, we have both foreground scribbles and background scribbles, and we can train our weakly supervised RGB-D saliency detection by concatenating RGB and depth in the input layer by adopting the model in [63]. Performance of the scribble annotation based baseline model is shown in Table 3.7 "Scribble".

Benchmark #3: Polygon based supervision. The polygon label is generated after majority voting. Fig. 3.3 (g) shows that the polygon label covers a larger area with better structure information than scribbles. We training directly with polygon annotations as pseudo labels, and provide performance of this baseline model in Table 3.7 "Polygon".

3.6 Conclusion

We proposed a novel complementary learning based RGB-D saliency detection framework that explicitly models complementary information between RGB images and depth data. By minimizing the mutual information between these two modes during training, our model focuses on the complementary information rather than the mutual information. In this fashion, our model is able to exploit the multi-mode information more effectively. Moreover, we introduced the largest RGB-D saliency detection dataset with five types of annotations to prosper the development of fully-/weakly-/un-supervised RGB-D saliency detection tasks. Extensive benchmarks on seven datasets and our new dataset demonstrate the superiority of our model compared to the existing 20 SOTAs. Note that, different from existing RGB or RGB-D based saliency training datasets [41, 167] where a single version of the ground truth

is provided to the input image or image pair, we provide multiple annotations for each input RGB-D image pair, making it convenient to perform prediction distribution estimation as our previous work in [60] in Chapter 2. In [60], the latent variable was introduced to model labeling variants, representing the "subjective nature" of saliency. As there existed no such rich annotations, [60] used a "hide and seek" technique [210] to generate diverse pseudo labels for each input image pair, which may be less accurate. With our diverse and accurate annotations for each input image pair, we can then perform accurate latent space exploring, which will be an extension of our work.

Chapter 4

Energy-Based Generative Cooperative Saliency Prediction

We introduced the latent variable model based RGB-D saliency detection frameworks in both Chapter 2 and Chapter 3. In this chapter, as an extension, we consider learning a conditional distribution over saliency maps, given an image, to model the uncertainty of predictions for RGB saliency detection. Specifically, we propose a generative saliency prediction model based on the conditional generative cooperative network, where a conditional latent variable model and a conditional energy-based model are jointly trained to predict saliency in a cooperative manner. The latent variable model serves as a coarse saliency model to produce a fast initial prediction, which is then refined by Langevin revision of the energy-based model that serves as a fine saliency model. We call this probabilistic coarse-to-fine saliency prediction. Moreover, we propose a cooperative learning while recovering strategy and apply it to weakly supervised saliency prediction, where saliency annotation of training images is partially observed. Lastly, we find that the learned energy function can serve as a refinement module that can refine the results of other pre-trained saliency prediction models. Experimental results show that our model can achieve state-of-the-art performance with stochastic predictions representing the uncertainty of model's predictions.

4.1 Introduction

Saliency detection aims to locate the region of an image that attracts human attention. Most existing saliency detection models [37, 33, 130, 34, 35, 46, 188] define saliency detection as a pixel-wise binary prediction task to achieve a deterministic mapping from the image to it's ground truth saliency map without modeling the "uncertainty" of predictions, representing the confidence of model's predictions. We argue that this may lead to overconfident models with poor generalization ability [211].

Further, the inherently subjective nature of saliency indicates that saliency detection is never a deterministic one-to-one mapping but a stochastic one-to-many mapping. We claim that instead of formulating saliency detection as a deterministic regression problem, it is more natural to represent the uncertainty of visual saliency

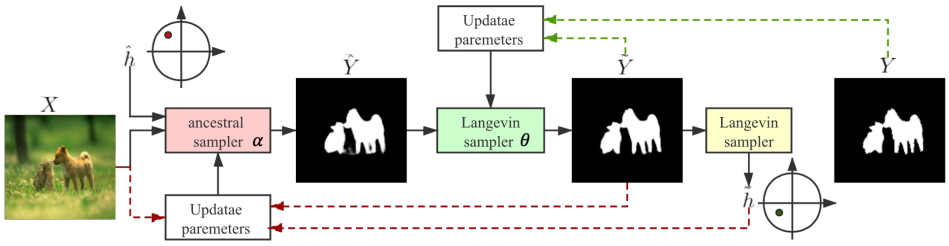


Figure 4.1: Probabilistic coarse to fine saliency prediction model. Training process: Image X and the latent variable $\hat{h}$ are fed to the "latent variable model" G_α to generate initial prediction $\hat{Y}$, which is then refined with the energy-based model U_θ to obtain $\tilde{Y}$. The refined prediction further updates the latent variable with gradient based MCMC. Testing process: The refined prediction $\tilde{Y}$ is our final prediction.

with a conditional probability distribution over saliency maps given an input image. To this end, developing probabilistic representational models for visual saliency is not only useful and natural for saliency prediction in computer vision, but also beneficial for understanding human attention in computational neuroscience.

With the above motivation, we propose a novel deep conditional generative model for probabilistic visual saliency prediction by representing the distribution of saliency outputs as a generative model conditioned on the input image, as shown in Fig. 4.1. The prediction process with such a model can be achieved by performing sampling from the distribution of outputs. Building upon recent developments in cooperative learning and sampling of deep energy-based models [56], we propose a conditional cooperative network for probabilistic visual saliency prediction.

The model consists of an energy-based model whose energy function is parameterized by a bottom-top neural networks and a conditional latent variable model whose transformation function is parameterized by an encoder-decoder framework. The energy-based model is flexible and powerful in distribution parameterization but computationally challenging because its maximum likelihood estimation typically requires Markov Chain Monte Carlo (MCMC) sampling to access the analytically intractable normalizing constant. By bringing in a latent variable model as an ancestral sampler to approximate or initialize the MCMC computational process for efficient sampling, the energy-based model can be learned efficiently. The energy function in the energy-based model, in turn, can be used to refine the latent variable model's samples, achieving the proposed "probabilistic coarse to fine saliency detection".

Moreover, based on the conditional cooperative network, we propose a novel *cooperative learning while cooperative recovering* strategy for weakly supervised saliency learning, in which we learn our model from incomplete data, where each training image is associated with a partially observed annotation (e.g., scribble [63]). At each learning iteration, the incomplete saliency ground truth is firstly recovered in the low-dimensional latent space of the latent variable model via inference, and then it is refined by pushing it to the local mode in the energy landscape of the energy-based model. The recovered saliency maps are treated as pseudo labels to train the pro-

posed framework. Although [60, 65] introduced probabilistic models in their framework, we introduce the first energy-based generative model for saliency prediction, which serves as a trainable loss function to improve model performance.

In experiments, we demonstrate that our model can not only achieve state-of-the-art performance in both fully supervised and weakly supervised saliency prediction, but also produce stochastic predictions representing uncertainty of model's prediction. Furthermore, we show that the learned energy function of the energy-based model can serve as a learned cost function to refine the results of other pre-trained saliency prediction models.

Our main contributions are fourfold. Firstly, we introduce the cooperative training based saliency detection framework to model the "uncertainty" of saliency with a latent variable model and an energy-based model. Secondly, we extend our method to saliency prediction with incomplete annotations to recover the unlabeled area. Thirdly, our energy-based model, once trained, can be easily embedded in existing saliency models as a refinement module. Lastly, experimental results in both fully and weakly supervised settings illustrate that we can achieve both high accuracy predictions and meaningful uncertainty estimation.

4.2 Related Work

Fully/Weakly Supervised Saliency Models. Existing fully supervised saliency prediction models [181, 31, 32, 34, 46, 33, 130, 37, 212, 35, 188] mainly focus on exploring image context information and generating structure-preserving predictions. [130, 35, 212, 181, 31, 46, 37] introduced saliency prediction models by effectively integrating higher- and lower-level features. [34, 32, 188] proposed edge-aware loss term to penalize errors along object boundaries. Note that all the above models are deterministic models. Recently, [60] introduced a conditional variational auto-encoder [50, 154] for stochastic RGB-D saliency detection. Similarly, we introduce a cooperative learning pipeline to achieve probabilistic coarse-to-fine RGB saliency detection via a latent variable model. However, [60] does not have a "fine" model, and this is the first time an energy-based model has been used for probalistic saliency detection.

The weakly supervised saliency models [41, 98, 99, 63] learn saliency from easy-to-obtain weak labels, including image-level labels [41, 98], noisy labels [99, 66, 124] or partial scribble labels [63]. Although a probabilistic model was explored in [65], they used a generative model for noise modeling. For the weakly-supervised task, we use a latent variable to model the distribution of the hidden clean saliency map.

Energy-Based Generative Models. Energy-based generative models [57, 87, 88, 89, 59, 90, 91, 92, 58, 93, 94, 95, 96] define an unnormalized density of a high-dimensional random variable of interest, which is in the form of the exponential of the negative energy function parameterized by a neural network. Maximum likelihood learning of the energy-based model typically requires MCMC sampling, which is computational challenging. To relieve the computational burden of MCMC, cooperative network in [56] proposes to learn a separate latent variable model (or directed graphical model)

to serve as an efficient approximate sampler for training the energy-based model. We propose a conditional model under the cooperative learning framework for visual saliency modeling and prediction. Our solution can be treated as an conditional version of [56]. While differently, we also extent the conditional model to weakly supervised learning with a cooperative learning while recovering algorithm. In this way, our model can learn the energy-based model from incomplete data for weakly supervised saliency prediction.

Conditional Deep Generative Models. Our framework belongs to the family of conditional generative models, which also include conditional generative adversarial networks (CGANs) [68] and conditional variational auto-encoders (CVAEs) [51]. Different from existing CGAN-based conditional generative models [80, 81, 82, 83, 84, 85, 86], which use GANs to detect higher-order inconsistency between ground truth and the prediction, or CVAEs based models [73, 60] in which a latent variable model representing an implicit density is learned, our model learns an explicit density via energy-based modeling. More importantly, our model allows an additional refinement for the latent variable model during prediction, which is sorely lacking in both CGANs and CVAEs frameworks.

4.3 Methodology

Our training dataset is $D = \{(X_i, Y_i)\}_{i=1}^n$, where n is size of the dataset. We propose a cooperative training [56] based RGB saliency detection method with a latent variable model and an energy-based model. The former uses an non-iterative ancestral sampler to generate an initial prediction with a latent variable h modeling uncertainty of saliency. The later refines the initial prediction with an iterative Langevin sampler incorporating higher-order structure disagreement, as shown in Fig. 4.1.

4.3.1 Probabilistic Saliency Prediction via Conditional Sampling

In this section, we propose a novel saliency prediction approach based on cooperative training of an energy-based model and a latent variable model.

Energy-Based Model as Fine Saliency Predictor. Let X be an image, and Y be its saliency map. The energy-based model $p_\theta(Y|X)$ defines a distribution of saliency Y given an image X by

$$p_\theta(Y|X) = \frac{p_\theta(Y, X)}{\int p_\theta(Y, X) dY} = \frac{1}{Z(X;\theta)} \exp[-U_\theta(Y, X)], \tag{4.1}$$

where the energy function $U_\theta(Y, X)$, parameterized by a bottom-up neural network, plays the role of a trainable objective function in the task of saliency prediction, and θ represents network parameter set. $Z(X;\theta) = \int \exp[-U_\theta(Y, X)] dY$ is the normalizing constant. When U_θ is learned and an image X is given, the prediction of saliency Y can be achieved by Langevin sampling [213] $Y \sim p_\theta(Y|X)$, which makes use of the

gradient of the energy function and iterates the following step:

$$Y_{\tau+1} = Y_\tau - \frac{\delta^2}{2}\frac{\partial U_\theta(Y_\tau, X)}{\partial Y} + \delta\Delta_\tau, \Delta_\tau \sim N(0, I_D), \tag{4.2}$$

where τ indexes the Langevin time steps, and δ is the step size. The Langevin dynamics [213] is equivalent to a stochastic gradient descent algorithm that seeks to find the minimum of the objective function defined by $U_\theta(Y, X)$. The Gaussian noise term Δ_τ is a Brownian motion that prevents the gradient descent from being trapped by local minima of $U_\theta(Y, X)$.

Latent Variable Model as Coarse Saliency Predictor. Let h be a latent Gaussian noise vector, $G_\alpha(X, h)$ be a mapping function parameterized by a noise-injected encoder-decoder network with skip connections. α contains all the learning parameters in the network. The latent variable model is given by:

$$h \sim N(0, I_d), Y = G_\alpha(X, h) + \epsilon, \epsilon \sim N(0, \sigma^2 I_D), \tag{4.3}$$

which defines an implicit conditional distribution of saliency Y given an image X, i.e., , $p_\alpha(Y|X) = \int p(h)p_\alpha(Y|X, h)dh$, where $p_\alpha(Y|X, h) = N(G_\alpha(X, h), \sigma^2 I_D)$. The saliency prediction can be achieved by an ancestral sampling by first sampling an injected Gaussian white noise h and then transforming the noise and the image X to a saliency map Y.

Saliency Prediction by Coarse-to-Fine Predictor. We propose to predict the saliency of an image by a cooperative sampling strategy, where we first use the coarse saliency predictor to generate an initial prediction $\hat{Y}$ via a non-iterative ancestral sampling, and then we use the fine saliency predictor to refine the initial prediction via iterative Langevin revision to obtain the revised saliency $\tilde{Y}$. We call this cooperative sampling based coarse-to-fine prediction. In this way, we take both advantages of these two saliency predictors in the sense that the fine saliency predictor (i.e., , Langevin sampler) is initialized by the efficient coarse saliency predictor (i.e., ., ancestral sampler), while the coarse saliency predictor is refined by the accurate fine saliency predictor that aims to minimize a cost function U_θ.

Since our conditional model represents a stochastic mapping, the prediction is stochastic as well. To evaluate the learned model on saliency prediction tasks, we can draw multiple h's from the prior $N(0, I_d)$ and use their average to generate $\hat{Y}$, then a Langevin dynamics with noise disabled (i.e., , gradient descent) is performed to push $\hat{Y}$ to its nearest local minimum $\tilde{Y}$ based on the learned energy function. The resulting $\tilde{Y}$ is treated as a prediction of our model.

4.3.2 Cooperative Learning of the Fine Saliency Predictor and the Coarse Saliency Predictor

MCMC-based Maximum Likelihood Estimation of Fine Saliency Predictor. Given a training dataset $\{(X_i, Y_i)\}_{i=1}^n$, we train the fine saliency predictor via maximum

likelihood estimation, which maximizes the log-likelihood of the data:

$$L(\theta) = \frac{1}{n}\sum_{i=1}^{n} \log p_\theta(Y_i|X_i), \tag{4.4}$$

whose gradient is:

$$\Delta\theta = \frac{1}{n}\sum_{i=1}^{n}\{E_{p_\theta(Y|X_i)}[\frac{\partial}{\partial\theta}U_\theta(Y, X_i)] - \frac{\partial}{\partial\theta}U_\theta(Y_i, X_i)\} \tag{4.5}$$

. We rely on the cooperative sampling in Eq. 4.2 to sample $\tilde{Y}_i \sim p_\theta(Y|X_i)$ to approximate the gradient:

$$\Delta\theta \approx \frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial\theta}U_\theta(\tilde{Y}_i, X_i) - \frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial\theta}U_\theta(Y_i, X_i). \tag{4.6}$$

We can use Adam with $\Delta\theta$ to update θ. We denote $\Delta\theta(\{Y_i\}, \{\tilde{Y}_i\})$ as a function of $\{Y_i\}$ and $\{\tilde{Y}_i\}$.

Maximum Likelihood Training of Coarse Saliency Predictor by MCMC Teaching. Even though the fine saliency predictor learns from the training data, the coarse saliency predictor learns to catch up with the fine saliency predictor by treating $\{(X, \tilde{Y})\}_{i=1}^n$ as training examples. The learning objective is to maximize the log-likelihood of the samples drawn from $p_\theta(Y|X)$, i.e., , $L(\alpha) = \frac{1}{n}\sum_{i=1}^n \log p_\alpha(\tilde{Y}_i|X_i)$, whose gradient can be computed by

$$\Delta\alpha = \sum_{i=1}^{n} E_{h\sim p_\alpha(h|Y_i,X_i)}\left[\frac{\partial}{\partial\alpha}\log p_\alpha(Y_i, h|X_i)\right]. \tag{4.7}$$

This leads to an MCMC-based solution that iterates (i) an inference step: inferring latent $\tilde{h}$ by sampling from posterior distribution $h \sim p_\alpha(h|Y, X)$ via Langevin dynamics, which iterates the following:

$$h_{\tau+1} = h_\tau + \frac{\delta^2}{2}\frac{\partial}{\partial h}\log p_\alpha(Y, h_\tau|X) + \delta\Delta_\tau, \tag{4.8}$$

where $\Delta_\tau \sim N(0, I_d)$, $\frac{\partial}{\partial h}\log p_\alpha(Y, h_\tau|X) = \frac{1}{\sigma^2}(Y - G_\alpha(X, h_\tau))\frac{\partial}{\partial h}G_\alpha(X, h_\tau) - h_\tau$, and (ii) a learning step: with $\{\tilde{h}_i, \tilde{Y}_i, X_i\}$, we update α via Adam optimizer with

$$\Delta\alpha \approx \frac{1}{n}\sum_{i=1}^{n}\frac{1}{\sigma^2}(\tilde{Y}_i - G_\alpha(X_i, \tilde{h}_i))\frac{\partial}{\partial\alpha}G_\alpha(X_i, \tilde{h}_i). \tag{4.9}$$

Since G_α is parameterized by a differentiable neural network, both $\frac{\partial}{\partial h}G_\alpha(X, h_\tau)$ in Equation (4.8) and $\frac{\partial}{\partial\alpha}G_\alpha(X_i, \tilde{h}_i)$ in Equation (4.9) can be efficiently computed by back-propagation. We denote $\Delta\alpha(\{\tilde{h}_i\}, \{\tilde{Y}_i\})$ as a function of $\{\tilde{h}_i\}$ and $\{\tilde{Y}_i\}$. Algorithm 2 presents a description of the cooperative learning algorithm with the fine and coarse saliency predictors.

4.3.3 Weakly Supervised Learning

Our model can be learned from a partially-observed training dataset $D' = \{(X_i, Y'_i)\}_{i=1}^n$, where only partial pixel-wise annotation is available, e.g., scribble [63].

Recovery by Coarse Saliency Predictor in Latent Space. Given an image X_i and its incomplete saliency map Y'_i, the recovery of the missing part of Y'_i can be achieved by first inferring the latent variable h'_i based on partially observed saliency information via $h'_i \sim p_\alpha(h|Y'_i, X_i)$, and then generating $\hat{Y}'_i = G_\alpha(X_i, h'_i)$ with the inferred h'_i. Let O_i be a binary mask, with the same size as Y', indicating the locations of visible annotations in Y'_i. O_i varies for different Y'_i and can be extracted from Y'_i. The Langevin dynamics for recovery iterates the same step in Equation (4.8) except that $\frac{\partial}{\partial h} \log p_\alpha(Y', h_\tau|X) = \frac{1}{\sigma^2}(O \circ (Y - G_\alpha(X, h_\tau))) \frac{\partial}{\partial h} G_\alpha(X, h_\tau) - h_\tau$, where sign $\circ$ denotes element-wise matrix multiplication operation.

Algorithm 2 Learning a coarse-to-fine saliency predictor for probabilistic saliency detection

Input:
(1) Training images $\{X_i\}_i^n$ with associated saliency maps $\{Y_i\}_i^n$;
(2) maximal number of learning iterations T.
Output: Parameters θ and α

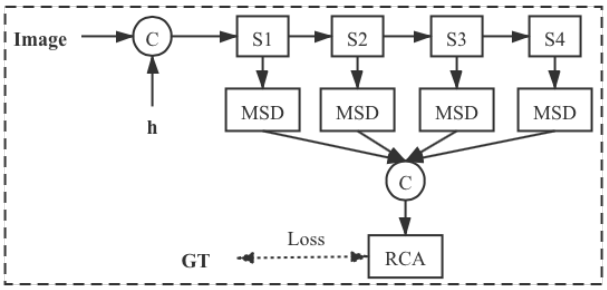
Initialize θ and α
for $t \leftarrow 1$ to T **do**
  Draw $\hat{h}_i \sim N(0, I_d)$
  Sample initial prediction $\hat{Y}_i = G_\alpha(X_i, \hat{h}_i)$.
  Revise $\hat{Y}_i$ to obtain $\tilde{Y}_i$ by Langevin in Eq. (4.2)
  Revise $\hat{h}_i$ to obtain $\tilde{h}_i$ by Langevin in Eq. (4.8)
  Update α with $\Delta\alpha(\{\tilde{h}_i\}, \{\tilde{Y}_i\})$ using Adam
  Update θ with $\Delta\theta(\{Y_i\}, \{\tilde{Y}_i\})$ using Adam
end for

Algorithm 3 Learning while recovering

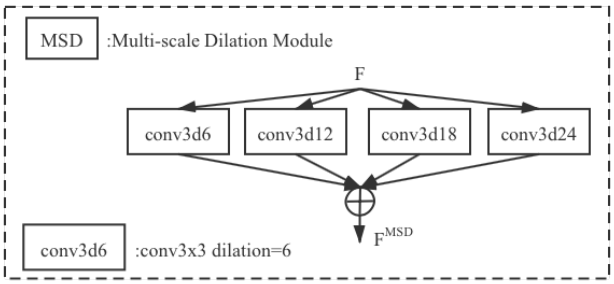
Input: (1) Images $\{X_i\}_i^n$ with incomplete annotations $\{Y'_i\}_i^n$; (2) Number of learning iterations T
Output: Parameters θ and α

Initialize θ and α
for $t \leftarrow 1$ to T **do**
  Infer $\hat{h}'_i$ from the visible part of Y'_i by Langevin dynamics in Eq. (4.8)
  Obtain initial recovery $\hat{Y}_i = G_\alpha(X_i, \hat{h}'_i)$.
  Revise $\hat{Y}'_i$ to obtain $\tilde{Y}'_i$ by Langevin in Eq. (4.2)
  Draw $\hat{h}_i \sim N(0, I_d)$
  Sample initial prediction $\hat{Y}_i = G_\alpha(X_i, \hat{h}_i)$.
  Revise $\hat{Y}_i$ to obtain $\tilde{Y}_i$ by Langevin in Eq. (4.2)
  Revise $\hat{h}_i$ to obtain $\tilde{h}_i$ by Langevin in Eq. (4.8)
  Update α with $\Delta\alpha(\{\tilde{h}_i\}, \{\tilde{Y}_i\})$ using Adam
  Update θ with $\Delta\theta(\{\tilde{Y}'_i\}, \{\tilde{Y}_i\})$ using Adam
end for

Recovery by Fine Saliency Predictor in Energy Landscape. With the initial re-



(a) The latent variable network



(b) Multi-scale dilation network

Figure 4.2: Network structure of the latent variable network, where $s1, ..., s4$ are four convolutional blocks of our backbone network, "RCA" is the residual channel attention module in [163].

covered result $\hat{Y}'$ by the coarse saliency predictor p_α, the fine saliency predictor p_θ can further recover the result by running finite steps of Langevin dynamics in Equation (4.2) initialized from $\hat{Y}'$ and obtain $\tilde{Y}'$. The underlying principle is that the initial recovery $\hat{Y}'$ might be just around the local modes of the energy function. A few steps of Langevin dynamics (i.e., , stochastic gradient descent) of p_θ, starting from $\hat{Y}'_i$, will push $\hat{Y}'_i$ to its nearby low energy mode, in which its potential fully observed version Y_i could be.

Cooperative Learning while Cooperative Recovering. At each iteration t, we perform the above cooperative recovery of the training saliency map $\{Y'\}_{i=1}^n$ via $p_{\theta^{(t)}}$ and $p_{\alpha^{(t)}}$, while learning $p_{\theta^{(t+1)}}$ and $p_{\alpha^{(t+1)}}$ from $\{X_i, \tilde{Y}_i'^{(t)}\}_{i=1}^n$, where $\tilde{Y}_i'^{(t)}$ is the recovered saliency map at t-th iteration. The parameter θ is still updated via Equation (4.6) except that we replace Y_i by $\tilde{Y}'_i$. That is, at each iteration, we use the recovered $\tilde{Y}'_i$, instead of the original Y_i, along with $\tilde{Y}_i$ to compute the gradient of log-likelihood, which is denoted by $\Delta\theta(\{\tilde{Y}'_i\}, \{\tilde{Y}_i\})$. The algorithm simultaneously performs (i) cooperative recovering of missing annotations; (ii) cooperative sampling to generate annotations; (iii) cooperative learning of the two models by updating parameters with both recovered annotations and generated annotations. See Algorithm 3 for a description of the learning while recovering algorithm.

4.3.4 Network Structure

Latent Variable Model: The latent variable model $G_\alpha(X, h)$ maps the concatenation of image X and latent variable h (we expand h to same spatial size of X) to coarse saliency map $\hat{Y}$ as shown in Fig. 4.2 (a). As indicated in existing saliency detection models [37, 33, 34, 130], two issues may influence the performance of saliency models: 1) structure recovery solution and 2) multi-scale strategy. To solve the first problem, we adopt structure aware similarity as in [34] to further penalize errors along edges of the prediction in training the latent variable model. To achieve multi-scale saliency detection, we design an encoder-decoder based network with ResNet50 [28] as backbone, which includes four convolutional group $s1, ..., s4$. We first design a multi-scale dilation network (MSD), shown in Fig. 4.2 (b), with gradually enlarged dilation rates to capture different scales of context information ("MSD" includes four 3×3 convolutional layers with dilation rates of $[3, 6, 12, 18]$ to map $\{s_c\}_{c=1}^4$ to an $M = 32$ channel feature map. Then the channel-reduced features are concatenated to form feature $\{r_c\}_{c=1}^4$ of each convolutional group). Then these channel-reduced features $\{r_c\}_{c=1}^4$ are again concatenated and fed to the "Residual Channel Attention" (RCA) module [163] to obtain discriminative feature representation $feat$. Finally, one 3×3 convolutional layer is adopted to map $feat$ to a one-channel saliency map $\hat{Y}$.
Energy Function: The energy function $U_\theta(Y, X)$ is composed of five convolutional layers and one fully connected layer. The kernel sizes, strides and output channel size of these five convolutional layers are (3,1,32), (4,2,64), (4,2,128), (4,2,256) and (4,1,1) respectively. Then, a fully connected layer is adopted to map the feature map after the last convolutional layer to a 100 dimensional feature vector, representing the energy of the input pair.

4.3.5 Alternative Generative Models for Saliency Detection

We further investigate alternative generative models for saliency detection, and design two extra stochastic RGB saliency detection frameworks by using CVAE [51] and CGAN [68].
Learning RGB Saliency via CVAE: CVAE [51] is a conditional directed graph model, which includes three variables, the input X or conditional variable that modulates the prior on Gaussian latent variable Z, that generate the output prediction Y. Two main modules are included in a conventional CVAE based framework: a generator model $P_\theta(Y|X, Z)$, which generates prediction Y with input X and Z as input, and an inference model $P_\theta(Z|X, Y)$, which infers the latent variable Z with input X and output Y as input. Learning a CVAE framework involves approximation of the true prior distribution $P_\theta(Z|X)$ with a inference model $Q_\phi(Z|X, Y)$. The parameter sets of a CVAE can be estimated in a stochastic variational Bayes (SGVB) [50] framework by maximizing the expected variational lower bound (ELBO) as:

$$L(\theta, \phi; X) = \mathbb{E}_{q_\phi(h|X,Y)}[\log(p_\theta(Y|X, h))] - D_{KL}(q_\phi(h|X, Y)||p_\theta(h|X)), \tag{4.10}$$

where the first term is the expected log-likelihood and the second term measures

the information lost using $Q_\phi(Z|X,Y)$ to approximate the true prior distribution of latent variable Z.

Following Eq. 4.10, we add an extra encoder $q_\phi(h|X,Y)$ and $p_\theta(h|X)$ to the generator model in Fig. 4.2 (a). The encoder is composed of five convolutional layers to map the input image X or concatenation of X and Y to low dimensional vectors: mean μ and standard deviation σ, and then the latent variable h is obtained with the reparameterazation trick as $z = \mu + \sigma \odot \epsilon$, where $\epsilon \in \mathcal{N}(0, I_d)$.

CGAN based Saliency Detection: Similar to CVAE, two different models (a generator and a discriminator) play the minimax game in CGAN as shown below:

$$\min_G \max_D V(D,G) = E_{(X,Y)\in p_{data}(X,Y)}[\log D(Y|X)] + E_{h\in q(h)}[\log(1 - D(G(X,h))], \tag{4.11}$$

where G and D are the generator model and discriminator model respectively, $p_{data}(X,Y)$ is the joint distribution of training data, $q(h)$ is the prior distribution of the latent variable h, which is usually defined as $q(h) = \mathcal{N}(0, I_d)$. We use the same generator as in Fig. 4.2 for generator of the CGAN. We design a fully convolutional discriminator as [85], to distinguish the prediction and ground truth pixel-wise as real or fake.

4.4 Experiments

Datasets: We used the DUTS dataset [41] for training the fully supervised model, and scribble annotation S-DUTS [63] for training the weakly supervised model. Testing images include 1) DUTS testing dataset, 2) ECSSD [189], 3) DUT [26], 4) HKU-IS [30], 5) THUR [190] and SOC [191].

Competing methods: We compared our method against eleven state-of-the-art fully supervised deep saliency detection methods: DGRL [181], PiCANet [31], F3Net [34], NLDF [32], PoolNet [46], BASNet [33], AFNet [40], MSNet [39], SCRN [130], ITSD [44] and LDF [188]. We also compare our weakly supervised solution in Section 4.3.3 with the scribble saliency detection model SSAL [63].

Evaluation Metrics: We evaluate performance of ours and competing methods with four saliency evaluation metrics, including: Mean Absolute Error ($\mathcal{M}$), mean F-measure (F_β), mean E-measure (E_ξ) [173] and S-measure (S_α) [172].

Training Details: We trained our model using Pytorch with a maximum of 30 epochs. Each image is rescaled to 352×352. ResNet50 [28] is chosen as backbone of the latent variable model. Empirically, we set the dimension of the latent space as $h = 8$. We used Adam with momentum 0.9 and decrease the learning rate 10% after 20 epochs. The learning rates of the latent variable model and EBM function are initialized to 1e-4 and 1e-3 respectively. It took 20 hours of training with batch size seven on a PC with an NVIDIA GeForce RTX GPU.

4.4.1 Comparison with Fully-supervised Models

Quantitative comparison: We evaluate performance of competing methods and ours and show results in Table 4.1, where "PCF" is the proposed **P**robabilistic **C**oarse-to-

Table 4.1: Performance comparison with benchmark saliency prediction models, including fully supervised models, weakly supervised models and alternative generator models.

	DUTS				ECSSD				DUT				HKU-IS				THUR				SOC			
Method	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$
	Deep Fully Supervised Models																							
DGRL	.846	.790	.887	.051	.902	.898	.934	.045	.809	.726	.845	.063	.897	.884	.939	.037	.816	.727	.838	.077	.791	.348	.820	.137
PiCAN	.842	.757	.853	.062	.898	.872	.909	.054	.817	.711	.823	.072	.895	.854	.910	.046	.818	.710	.821	.084	.801	.332	.810	.133
F3Net	.888	.852	.920	.035	.919	.921	.943	.036	.839	.766	.864	.053	.917	.910	.952	.028	.838	.761	.858	.066	.828	.340	.846	.098
NLDF	.816	.757	.851	.065	.870	.871	.896	.066	.770	.683	.798	.080	.879	.871	.914	.048	.801	.711	.827	.081	.816	.319	.837	.106
PoolN	.887	.840	.910	.037	.919	.913	.938	.038	.831	.748	.848	.054	.919	.903	.945	.030	.834	.745	.850	.070	.829	.355	.846	.098
BASN	.876	.823	.896	.048	.910	.913	.938	.040	.836	.767	.865	.057	.909	.903	.943	.032	.823	.737	.841	.073	**.841**	.359	.864	**.092**
AFNet	.867	.812	.893	.046	.907	.901	.929	.045	.826	.743	.846	.057	.905	.888	.934	.036	.825	.733	.840	.072	.700	.062	.684	.115
MSNet	.862	.792	.883	.049	.905	.886	.922	.048	.809	.710	.831	.064	.907	.878	.930	.039	.819	.718	.829	.079	-	-	-	-
SCRN	.885	.833	.900	.040	.920	.910	.933	.041	.837	.749	.847	.056	.916	.894	.935	.034	.845	.758	.858	.066	.838	.363	.859	.099
ITSD	.885	.840	.913	.041	.919	.917	.941	.037	.840	.768	.865	.061	.917	.904	.947	.031	.836	.753	.852	.070	.773	.361	.792	.166
LDF	.892	.861	.925	.034	.919	.923	.943	.036	.839	.770	.865	.052	.920	.913	.953	.028	.842	.768	.863	.064	.835	**.369**	.856	.103
PCF	**.890**	**.856**	**.924**	**.034**	**.926**	**.930**	**.954**	**.031**	**.852**	**.788**	**.879**	**.046**	**.923**	**.917**	**.957**	**.026**	**.847**	**.771**	**.867**	**.061**	.839	.368	**.860**	**.092**
	Weakly Supervised Models																							
SSAL	.803	.747	.865	.062	.863	.865	.908	.061	.785	.702	.835	.068	.865	.858	.923	.047	.800	**.718**	.837	.077	.804	.309	.793	.143
WPCF	**.813**	**.755**	**.863**	**.059**	**.872**	**.874**	**.910**	**.060**	**.791**	**.707**	**.840**	**.061**	**.871**	**.859**	**.929**	**.042**	**.804**	.717	**.839**	**.074**	**.812**	**.314**	**.806**	**.137**
	Alternative Generator Models																							
CVAE	.866	.824	.900	.041	.906	.910	.932	.043	.816	.737	.844	.055	.910	.903	.943	.032	.835	.755	.859	**.065**	**.843**	.361	**.866**	.098
CGAN	.846	.785	.883	.049	.900	.895	.928	.047	.799	.705	.828	.063	.894	.875	.930	.039	.823	.732	.850	.071	**.841**	.362	.859	.103

Fine fully supervised saliency model. We observe consistent performance improvement of "PCF" compared with benchmark models. We also designed two alternative generator network based saliency detection pipelines with CVAEs [51] and CGANs [68] respectively (details about these two alternative network are introduced Section 4.3.5), and performance is shown as "CVAE" and "CGAN". As indicated in both Table 4.1 and our experience of training, we found that the CGANs-based model is very sensitive to the weights of the discriminator loss, and the whole training is not very stable. The CVAEs-based model can achieve stable training, while imbalanced training (generator model and inference model) may lead to posterior collapse [141], where the latent variable h is independent of the prediction Y, thus it fails to capture the uncertainty of human annotation. The performance gap between ours and alternative generator network further illustrates superior performance of our solution.

Qualitative comparison: As a generative model, we intend to model human uncertainty of annotation, thus, diversity of prediction is a main standard to evaluate performance of our model. We visualize predictions of ours and two competing methods (F3Net and SCRN) as shown in Fig. 4.3, where "Our Samples" represent our predictions with four iterations of sampling, and "Ours" is computed with an average of multiple latent variable h as input (introduced in Section 4.3.1). Fig. 4.3 shows that the deterministic one-to-one mapping may over-confidently segment too many or too few regions as salient. The proposed solution can produce multiple predictions with each iteration of sampling, which is more consistent with the "subjective" nature of saliency. Furthermore, as shown in the "Ours" column, when

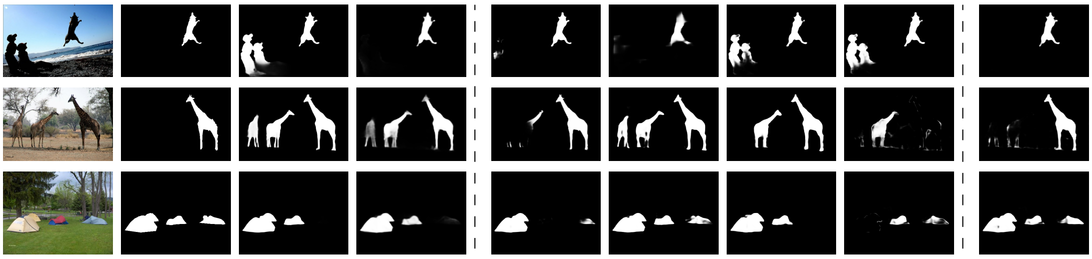


Figure 4.3: Visual comparison of predictions of ours and competing saliency models.

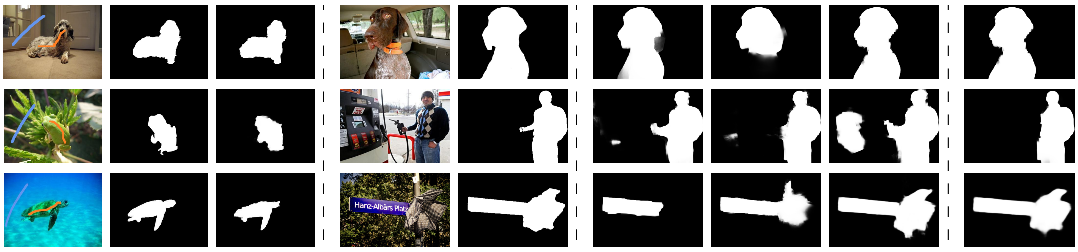


Figure 4.4: Examples showing the training and testing related data of our "cooperative learning while recovering" weakly supervised learning solution. Note that, scribble is only used during training stage.

Table 4.2: Performance comparison of extra module analysis models.

	DUTS				ECSSD				DUT				HKU-IS				THUR				SOC			
	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$
	EBM as Refinement Module																							
NLDF_R	.867	.827	.911	.040	.906	.911	.939	.041	.814	.738	.850	.058	.909	.903	.951	.030	.827	.747	.854	.068	.812	.327	.825	.132
PoolN_R	.890	.852	.921	.035	.923	.920	.945	.036	.840	.766	.867	.050	.925	.914	.952	.027	.834	.745	.850	.070	.829	.355	.846	.106
BASN_R	.882	.836	.893	.042	.923	.919	.944	.035	.843	.788	.875	.050	.913	.914	.950	.030	.834	.745	.850	.071	.841	.359	.864	.092
SCRN_R	.887	.854	.922	.035	.923	.923	.944	.036	.839	.759	.861	.052	.919	.913	.953	.028	.849	.763	.863	.064	.846	.369	.868	.091
	Ablation Study																							
DGen	.872	.832	.917	.038	.909	.911	.940	.040	.820	.745	.857	.055	.911	.903	.952	.030	.828	.748	.855	.068	.832	.359	.852	.101
SGen	.881	.849	.920	.036	.918	.923	.946	.036	.834	.766	.865	.052	.918	.914	.954	.027	.833	.756	.856	.068	.835	.362	.864	.095
DEBM	.883	.851	.921	.035	.916	.920	.944	.036	.835	.769	.866	.050	.919	.915	.955	.026	.833	.757	.858	.064	.841	.367	.870	.096
CRFs	.880	.842	.925	.036	.905	.909	.931	.042	.826	.739	.853	.056	.910	.914	.957	.031	.837	.769	.862	.066	.842	.363	.865	.094
Ours	**.890**	**.856**	**.924**	**.034**	**.926**	**.930**	**.954**	**.031**	**.852**	**.788**	**.879**	**.046**	**.923**	**.917**	**.957**	**.026**	**.847**	**.771**	**.867**	**.061**	.839	.368	**.860**	**.092**

taking the average of latent variable h as input, we can produce predictions that are most similar to the provided ground truth compared with the competing methods.

4.4.2 Weakly Supervised Saliency Detection

We extend our solution to weakly supervised saliency detection with scribble annotation [63]. As shown in Fig. 4.4 (a), compared with the fully annotated ground truth ("GT"), the scribble annotation ("Scribble") is sparse (orange scribble is salient

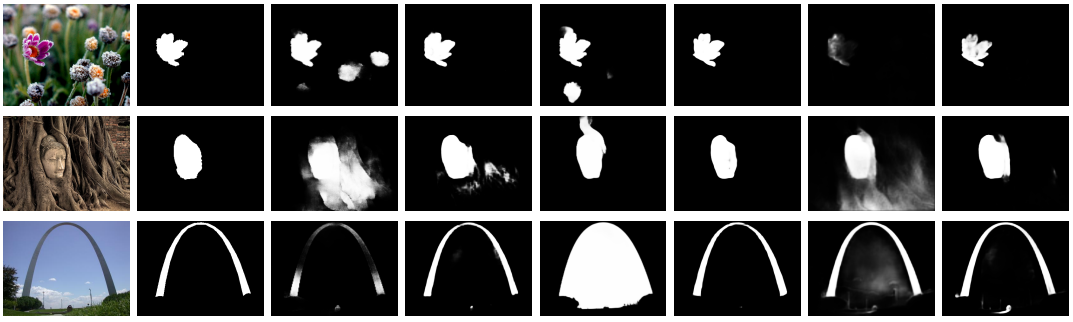


Figure 4.5: Visual comparison of without ("NLDF") and with ("NLDF_R") EBM as refinement module.

region, and blue scribble represents background). With the proposed "cooperative learning while recovering" strategy, we can recover the missing annotation region as shown in "Rec-ed". Different from [63], which incorporates an extra edge detection module and smoothness loss [109] to push the prediction share similar shape as the input image, we instead put the smoothness constraint into the energy-based model while updating the parameter set θ in Eq. 4.6. "WPCF" in Table 4.1 is our weakly supervised learning model. Note that, the scribble annotation is only used during training as partial supervision. During testing, given input image, we can still produce stochastic predictions as shown in Fig. 4.4 (b). Compared with "SSAL", "WPCF" achieves consistent better performance. Moreover, as a probabilistic model, "WPCF" can generate multiple predictions representing the uncertainty of human annotation, while "SSAL" can not.

4.4.3 Energy Function as a Refinement Module

As shown in Eq. 4.2, the energy-based model can iteratively refine the prediction with Langevin sampling. With the trained EBM, we treat it as a refinement module, and add it to four existing deep saliency detection models. Performance is shown in Table 4.2 and Fig. 4.5. "NLDF_R", "SCRN_R" "BASN_R" and "PoolN_R", representing using EBM to refine NLDF [32], SCRN [130], BASN [33] and PoolN [46] respectively[1]. Compared with the original performance of the above four existing models, we observe consistent performance improvement with EBM as a refinement module, especially for NLDF, we achieve around 5% performance improvement for S-measure, F-measure and E-measure, and 2% performance improvement for MAE. We further show three examples of those models with and without EBM as refinement. Both performance improvement and visual comparison illustrates the effectiveness of EBM as a refinement module.

[1]We choose these four models due to their accessible codes and saliency maps

4.4.4 Ablation Study

We carried out the following experiments as shown in Table 4.2 to further analyse our solution.

Training the deterministic latent variable model: G_α can be trained directly without the latent variable h, i.e., $G_\alpha(X)$, and the result is shown as "DGen". We observe competing performance of "DGen" compared with existing benchmark fully supervised saliency detection models. While, the gap between "DGen" and ours "Ours" illustrates superior performance of the proposed solution.

Training directly the latent variable model: We treat the stochastic saliency encoder-decoder $G_\alpha(X, h)$ as our final model, where the latent variable h is updated through Langevin dynamics as shown in Eq. 4.8. The result is shown in "SGen". Compared with "DGen", "SGen" achieves better performance, which illustrates effectiveness of the latent variable model.

Training the deterministic latent variable model with EBM as refinement: We add EBM to "DGen" as a refinement module during both training and testing. The result is shown in "DEBM". The gap between "DEBM" and "DGen" is consistent with EBM as refinement of pre-trained saliency models in Section 4.4.3, which further proves feasibility of EBM as saliency refinement module.

Refine the latent variable model with CRFs [29]: As a popular post-processing technique, dense CRFs [29] can be adopted to refine predictions of our latent variable model. We refine "DGen" with the dense CRFs, where we use the same hyper parameters of [29] for semantic segmentation. The results are shown as "CRFs". We observe unstable performance of "CRFs", and the main reason for this is the difficulty in finding the effective hyper parameters. In order to find the right hyper parameter set, [214] introduced the trainable CRFs, which is just similar to our trainable EBM.

4.5 Conclusion

In this chapter, we propose a generative saliency prediction model based on the conditional generative cooperative network, where a latent variable model and an energy-based model are jointly trained in a cooperative learning scheme. The latent variable model serves as a coarse saliency predictor that provides a fast initial prediction, which is then refined by Langevin revision of the energy-based model. Moreover, we introduce a cooperative learning while recovering strategy and extend our model to weakly supervised saliency detection. Further, we find that the learned energy function can serves as a refinement module, which can be easily embedded into existing pre-trained saliency models. Experimental results compared with both fully supervised and weakly supervised saliency models illustrate the effectiveness of the proposed framework. To focus on the learning pipeline, we discuss RGB saliency detection to avoid analysing the multi-modal learning issue within the RGB-D saliency detection framework of Chapter 2 and Chapter 3. However, our learning pipeline works in general, which can be easily extend to existing RGB-D saliency framework. We will work on it to further evaluate our learning pipeline.

Chapter 5

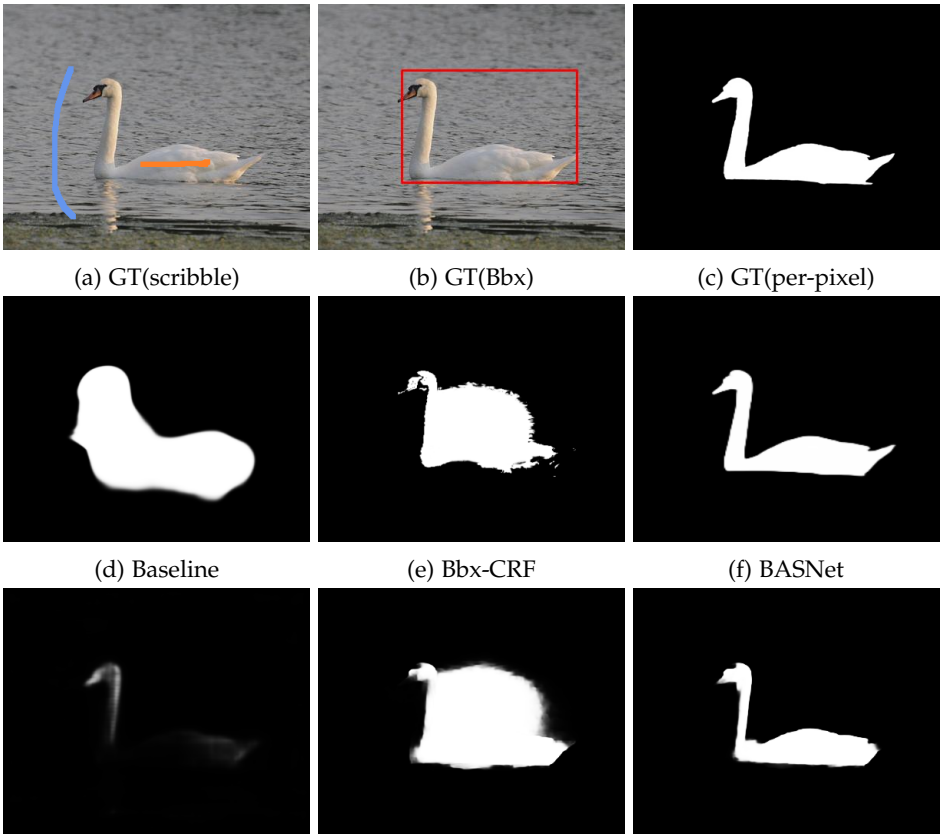
Weakly-Supervised Salient Object Detection via Scribble Annotations

For both RGB-D saliency detection models in Chapter 2 and 3, and RGB saliency detection framework in Chapter 4, we followed the supervised learning pipeline, with pixel-wise clean ground truth saliency map as supervision. Compared with laborious pixel-wise dense labeling, it is much easier to label data by scribbles, which only costs 1∼2 seconds to label one image. However, using scribble labels to learn salient object detection has not been explored. In this chapter, we propose a weakly-supervised salient object detection model to learn saliency from such annotations. In doing so, we first relabel an existing large-scale salient object detection dataset with scribbles, namely S-DUTS dataset. Since object structure and detail information is not identified by scribbles, directly training with scribble labels will lead to saliency maps of poor boundary localization. To mitigate this problem, we propose an auxiliary edge detection task to localize object edges explicitly, and a gated structure-aware loss to place constraints on the scope of structure to be recovered. Moreover, we design a scribble boosting scheme to iteratively consolidate our scribble annotations, which are then employed as supervision to learn high-quality saliency maps. As existing saliency evaluation metrics neglect to measure structure alignment of the predictions, the saliency map ranking metric may not comply with human perception. We present a new metric, termed saliency structure measure, as a complementary metric to evaluate sharpness of the prediction. Extensive experiments on six benchmark datasets demonstrate that our method not only outperforms existing weakly-supervised/unsupervised methods, but also is on par with several fully-supervised state-of-the-art models[1].

5.1 Introduction

Visual salient object detection (SOD) aims at locating interesting regions that attract human attention most in an image. Conventional salient object detection methods [24, 23] based on hand-crafted features or human experience may fail to obtain high-quality saliency maps in complicated scenarios. The deep learning based salient object detection models [130, 60] have been widely studied, and significantly boost

[1]Our code and data is publicly available at: https://github.com/JingZhang617/Scribble_Saliency.



(a) GT(scribble) (b) GT(Bbx) (c) GT(per-pixel)

(d) Baseline (e) Bbx-CRF (f) BASNet

(g) WSS (h) Bbx-Pred (i) Ours

Figure 5.1: (a) Our scribble annotations. (b) Ground-truth bounding box. (c) Ground-truth pixel-wise annotations. (d) Baseline model: trained directly on scribbles. (e) Refined bounding box annotation by DenseCRF [29]. (f) Result of a fully-supervised SOD method [33]. (g) Result of model trained on image-level annotations [41] (h) Model trained on the annotation (e). (i) Our result.

the saliency detection performance. However, these methods highly rely on a large amount of labeled data, which require time-consuming and laborious pixel-wise annotations. To achieve a trade-off between labeling efficiency and model performance, several weakly supervised or unsupervised methods [98, 102, 99, 66] have been proposed to learn saliency from sparse labeled data [98, 102] or infer the latent saliency from noisy annotations [99, 66].

We propose a new weakly-supervised salient object detection framework by learning from low-cost labeled data, (i.e., , scribbles, as seen in Fig. 5.1(a)). Here, we opt to scribble annotations because of their flexibility (although bounding box annotation is an option, it's not suitable for labeling winding objects, thus leading to inferior saliency maps, as seen in Fig. 5.1 (h)). Since scribble annotations are usually very sparse, object structure and details cannot be easily inferred. Directly training a deep model with sparse scribbles by partial cross-entropy loss [111] may lead to saliency maps of poor boundary localization, as illustrated in Fig. 5.1 (d).

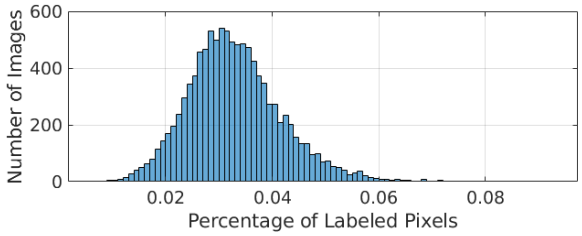


Figure 5.2: Percentage of labeled pixels in the S-DUTS dataset.

To achieve high-quality saliency maps, we present an auxiliary edge detection network and a gated structure-aware loss to enforce boundaries of our predicted saliency map to align with image edges in the salient region. The edge detection network forces the network to produce feature highlight object structure, and the gated structure-aware loss allows our network to focus on the salient region while ignoring the structure of the background. We further develop a scribble boosting manner to update our scribble annotations by propagating the labels to larger receptive fields of high confidence. In this way, we can obtain denser annotations as shown in Fig. 5.7 (g).

Due to the lack of scribble based saliency datasets, we relabel an existing saliency training dataset DUTS [41] with scribbles, namely S-DUTS dataset, to verify our method. DUTS is a widely used salient object detection dataset, which contains 10,553 training images. Annotators are asked to scribble the DUTS dataset according to their first impressions without showing them the ground-truth salient objects. Fig. 5.2 indicates the percentage of labeled pixels across the whole S-DUTS dataset. On average, around 3% of the pixels are labeled (either foreground or background) and the others are left as unknown pixels, demonstrating that the scribble annotations are very sparse. Note that, we only use scribble annotation as supervision signal during training, and we take RGB image as input to produce dense saliency map during testing.

Moreover, the rankings of saliency maps based on traditional mean absolute error (MAE) may not comply with human visual perception. For instance, in the 1^{st} row of Fig. 5.3, the last saliency map is visually better than the fourth one and the third one is better than the second one. We propose saliency structure measure (B_μ) as a complementary metric of existing evaluation metrics that takes the structure alignment of the saliency map into account. The measurements based on B_μ are more consistent with human perception, as shown in the 2^{nd} row of Fig. 5.3.

We summarize our main contributions as: (1) we present a new weakly-supervised salient object detection method by learning saliency from scribbles, and introduce a new scribble based saliency dataset S-DUTS; (2) we propose a gated structure-aware loss to constrain a predicted saliency map to share similar structure with the input image in the salient region; (3) we design a scribble boosting scheme to expand our scribble annotations, thus facilitating high-quality saliency map acquisition; (4) we present a new evaluation metric to measure the structure alignment of predicted saliency maps, which is more consistent with human visual perception; (5)

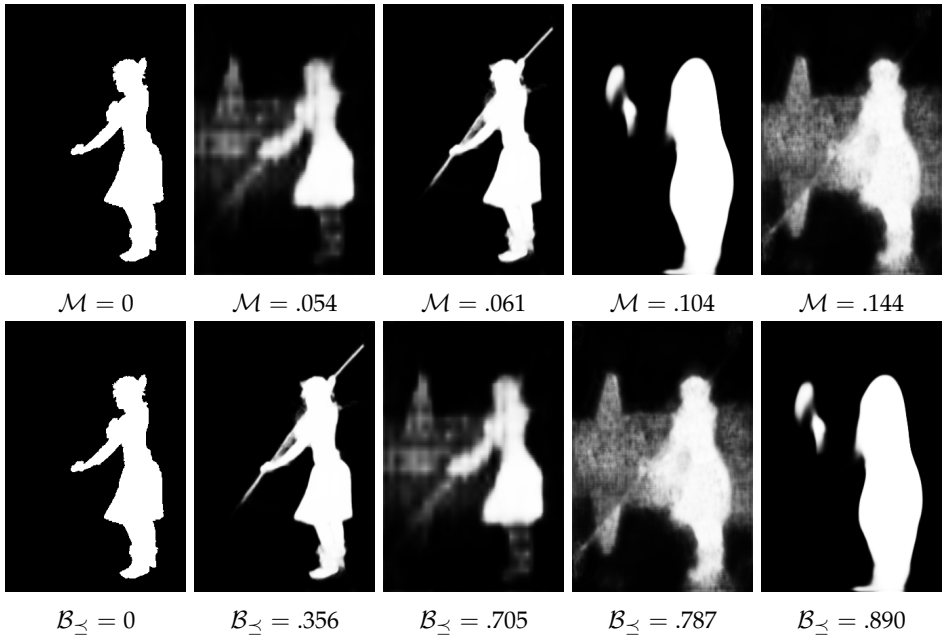


Figure 5.3: Saliency map ranking based on Mean Absolute Error (1^{st} row) and our proposed Saliency Structure Measure (2^{nd} row).

experimental results on six salient object detection benchmarks demonstrate that our method outperforms state-of-the-art weakly-supervised algorithms.

5.2 Related Work

Deep fully supervised saliency detection models [33, 215, 130, 60, 216, 35, 217] have been widely studied. As our method is weakly supervised, we mainly discuss related weakly-supervised dense prediction models and approaches to recover detail information from weak annotations.

5.2.1 Learning Saliency from Weak Annotations

To avoid requiring accurate pixel-wise labels, some SOD methods attempt to learn saliency from low-cost annotations, such as bounding boxes [218], image-level labels [41, 98], and noisy labels [66, 124, 99], *etc.*. This motivates SOD to be formulated as a weakly-supervised or unsupervised task. Wang *et al.* [41] introduced a foreground inference network to produce saliency maps with image-level labels. With the same weak labels, Hsu *et al.* [101] presented a category-driven map generator to learn saliency from class activation map. Li *et al.* [98] adopted an iterative learning strategy to update an initial saliency map generated from unsupervised saliency methods by learning with image-level supervision. A fully connected CRF [29] was utilized in [41, 98] as post-processing to refine the produced saliency map. Zeng *et al.* [102] proposed to train saliency models with diverse weak supervision sources, includ-

ing category labels, captions, and unlabeled data. Zhang *et al.* [124] fused saliency maps from unsupervised methods with heuristics within a deep learning framework. In a similar setting, Zhang *et al.* [66] proposed to collaboratively update a saliency prediction module and a noise module to learn a saliency map from multiple noisy labels.

5.2.2 Weakly-Supervised Semantic Segmentation

Dai *et al.* [103] and Khoreva [104] proposed to learn semantic segmentation from bounding boxes in a weakly-supervised way. Hung *et al.* [85] randomly interleaved labeled and unlabeled data, and trained a network with an adversarial loss on the unlabeled data for semi-supervised semantic segmentation. Shi *et al.* [105] tackled the weakly-supervised semantic segmentation problem by using multiple dilated convolutional blocks of different dilation rates to encode dense object localization. Li *et al.* [106] presented an iterative bottom-up and top-down semantic segmentation framework to alternatingly expand object regions and optimize segmentation network with image tag supervision. Huang *et al.* [107] introduced a seeded region growing technique to learn semantic segmentation with image-level labels. Vernaza *et al.* [108] designed a random walk based label propagation method to learn semantic segmentation from sparse annotations.

5.2.3 Recovering Structure from Weak Labels

As weak annotations do not contain complete semantic region of the specific object, the predicted object structure is often incomplete. To preserve rich and fine-detailed semantic information, additional regularizations are often employed. Two main solutions are widely studied, including graph model based methods (e.g., CRF [29]) and boundary based losses [110]. Tang *et al.* [111] introduced a normalized cut loss as a regularizer with partial cross-entropy loss for weakly-supervised image segmentation. Tang *et al.* [219] modeled standard regularizers into a loss function over partial observation for semantic segmentation. Obukhov *et al.* [112] proposed a gated CRF loss for weakly-supervised semantic segmentation. Lampert *et al.* [110] introduced a constrain-to-boundary principle to recover detail information for weakly-supervised image segmentation.

5.2.4 Comparison with Existing Scribble Models

Although scribble annotations have been used in weakly-supervised semantic segmentation [220, 221], our proposed scribble based salient object detection method is different from them in the following aspects: (1) semantic segmentation methods target at class-specific objects. In this manner, class-specific similarity can be explored. On the contrary, salient object detection does not focus on class-specific objects, thus object category related information is not available. For instance, a leaf can be a salient object while the class category is not available in the widely used image-level label dataset [222, 223]. Therefore, we propose edge-guided gated structure-aware

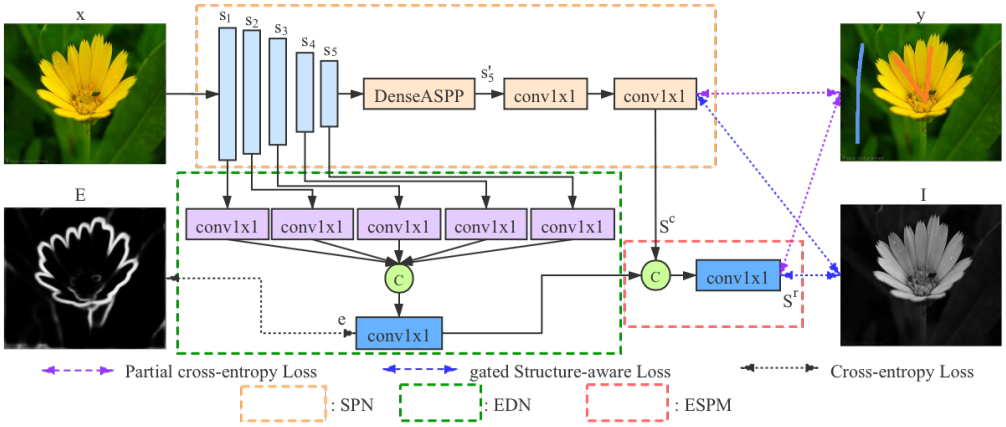


Figure 5.4: Illustration of our network. For simplicity, we do not show the scribble boosting mechanism here. "I" is the intensity image of input "x". "C": concatenation operation; "conv1x1": 1×1 convolutional layer.

loss to obtain structure information from image instead of depending on image category. (2) although boundary information has been used in [221] to propagate labels, Wang *et al.* [221] regressed boundaries by an ℓ_2 loss. Thus, the structure of the segmentation may not be well aligned with the image edges. In contrast, our method minimizes the differences between first order derivatives of saliency maps and images, and leads to saliency map better aligned with image structure. (3) benefiting from our developed boosting method and the intrinsic property of salient objects, our method requires only scribble on any salient region as shown in Fig. 5.9, while scribbles are required to traverse all those semantic categories for scribble based semantic segmentation [220, 221].

5.3 Learning Saliency from Scribbles

Let's define our training dataset as: $D = \{x_i, y_i\}_{i=1}^{N}$, where x_i is an input image, y_i is its corresponding annotation, N is the size of the training dataset. For fully-supervised salient object detection, y_i is a pixel-wise label with 1 representing salient foreground and 0 denoting background. We define a new weakly-supervised saliency learning problem from scribble annotations, where y_i in our case is scribble annotation used during training, which includes three categories of supervision signal: 1 as foreground, 2 as background and 0 as unknown pixels. In Fig. 5.2, we show the percentage of annotated pixels of the training dataset, which indicates that around 3% of pixels are labeled as foreground or background in our scribble annotation.

There are three main components in our network, as illustrated in Fig. 5.4: (1) a saliency prediction network (SPN) to generate a coarse saliency map s^c, which is trained on scribble annotations by a partial cross-entropy loss [111]; (2) an edge detection network (EDN) is proposed to enhance structure of s^c, with a gated structure-

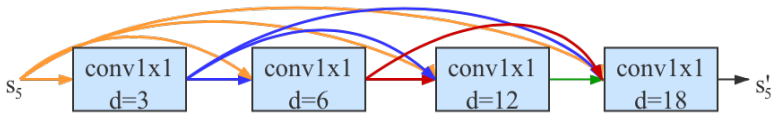


Figure 5.5: Our "DenseASPP" module. "conv1x1 d=3" represents a 1×1 convolutional layer with a dilation rate 3.

aware loss employed to force the boundaries of saliency maps to comply with image edges; (3) an edge-enhanced saliency prediction module (ESPM) is designed to further refine the saliency maps generated from SPN.

5.3.1 Weakly-Supervised Salient Object Detection

Saliency prediction network (SPN): We build our front-end saliency prediction network based on VGG16-Net [67] by removing layers after the fifth pooling layer. Similar to [224], we group the convolutional layers that generate feature maps of the same resolution as a stage of the network (as shown in Fig. 5.4). Thus, we denote the front-end model as $f_1(x,\theta) = \{s_1,...,s_5\}$, where $s_m(m = 1,...,5)$ represents features from the last convolutional layer in the m-th stage ("relu1_2, relu2_2, relu3_3, relu4_3, relu5_3"), θ is the front-end network parameters.

As discussed in [105], enlarging receptive fields by different dilation rates can propagate the discriminative information to non-discriminative object regions. We employ a dense atrous spatial pyramid pooling (DenseASPP) module [162] on top of the front-end model to generate feature maps s'_5 with larger receptive fields from feature s_5. In particular, we use varying dilation rates in the convolutional layers of DenseASPP. Then, two extra 1×1 convolutional layers are used to map s'_5 to a one channel coarse saliency map s^c.

As we have unknown category pixels in the scribble annotations, partial cross-entropy loss [111] is adopted to train our SPN:

$$\mathcal{L}_s = \sum_{(u,v)\in J_l} \mathcal{L}_{u,v}, \tag{5.1}$$

where J_l represents the labeled pixel set, (u,v) is the pixel coordinates, and $\mathcal{L}_{u,v}$ is the cross-entropy loss at (u,v).

Edge detection network (EDN): Edge detection network encourages SPN to produce saliency features with rich structure information. We use features from the intermediate layers of SPN to produce one channel edge map e. Specifically, we map each $s_i(i = 1,...,5)$ to a feature map of channel size M with a 1×1 convolutional layer. Then we concatenate these five feature maps and feed them to a 1×1 convolutional layer to produce an edge map e. A cross-entropy loss $\mathcal{L}_e$ is used to train EDN:

$$\mathcal{L}_e = \sum_{u,v}(E \log e + (1 - E) \log(1 - e)), \tag{5.2}$$

where E is pre-computed by an existing edge detector [225].

Edge-enhanced saliency prediction module (ESPM): We introduce an edge-enhanced saliency prediction module to refine the coarse saliency map s^c from SPN and obtain an edge-preserving refined saliency map s^r. Specifically, we concatenate s^c and e and then feed them to a 1×1 convolutional layer to produce a saliency map s^r. Note that, we use the saliency map s^r as the final output of our network. Similar to training SPN, we employ a partial cross-entropy loss with scribble annotations to supervise s^r.

Gated structure-aware loss: Although ESPM encourages the network to produce saliency map with rich structure, there exists no constraints on scope of structure to be recovered. Following the "Constrain-to-boundary" principle [110], we propose a gated structure-aware loss, which encourages the structure of a predicted saliency map to be similar to the salient region of an image.

We expect the predicted saliency map having consistent intensities inside the salient region and distinct boundaries at the object edges. Inspired by the smoothness loss [165, 109], we also impose such constraint inside the salient regions. Recall that the smoothness loss is developed to enforce smoothness while preserving image structure across the whole image region. However, salient object detection intends to suppress the structure information outside the salient regions. Therefore, enforcing the smoothness loss across the entire image regions will make the saliency prediction ambiguous, as shown in Tabel 5.2 "M3".

To mitigate this ambiguity, we employ a gate mechanism to let our network focus on salient regions only to reduce distraction caused by background structure. Specifically, we define the gated structure-aware loss as:

$$\mathcal{L}_b = \sum_{u,v} \sum_{d \in \overrightarrow{x}, \overrightarrow{y}} \Psi(|\partial_d s_{u,v}| e^{-\alpha |\partial_d (G \cdot I_{u,v})|}), \tag{5.3}$$

where Ψ is defined as $\Psi(s) = \sqrt{s^2 + 1e^{-6}}$ to avoid calculating the square root of zero, $I_{u,v}$ is the image intensity value at pixel (u, v), d indicates the partial derivatives on the $\overrightarrow{x}$ and $\overrightarrow{y}$ directions, and G is the gate for the structure-aware loss (see Fig .5.6 (d)). The gated structure-aware loss applies L1 penalty on gradients of saliency map s to encourages it to be locally smooth, with an edge-aware term ∂I as weight to maintain saliency distinction along image edges.

Specifically, as shown in Fig. 5.6, with predicted saliency map (a)) during training, we dilate it with a square kernel of size $k = 11$ to obtain an enlarged foreground region (c)). Then we define gate (d)) as binarized (c)) by adaptive thresholding. As seen in Fig. 5.6(e), our method is able to focus on the saliency region and predict sharp boundaries in a saliency map.

Objective Function: As shown in Fig. 5.4, we employ both partial cross-entropy loss $\mathcal{L}_s$ and gated structure-aware loss $\mathcal{L}_b$ to coarse saliency map s^c and refined map s^r, and use cross-entropy loss $\mathcal{L}_e$ for the edge detection network. Our final loss

(a) (b) (c) (d) (e)

Figure 5.6: Gated structure-aware constraint: (a) Initial predicted saliency map. (b) Image edge map. (c) Dilated version of (a). (d) Gated mask in Eq. 5.3. (e) Gated edge map.

function is then defined as:

$$\mathcal{L} = \mathcal{L}_s(s^c, y) + \mathcal{L}_s(s^r, y) + \beta_1 \cdot \mathcal{L}_b(s^c, x) + \beta_2 \cdot \mathcal{L}_b(s^r, x) + \beta_3 \cdot \mathcal{L}_e, \tag{5.4}$$

where y indicates scribble annotations. The partial cross-entropy loss $\mathcal{L}_s$ takes scribble annotation as supervision, while gated structure-aware loss $\mathcal{L}_b$ leverages image boundary information. These two losses do not contradict each other since $\mathcal{L}_s$ focuses on propagating the annotated scribble pixels to the foreground regions (relying on SPN), while $\mathcal{L}_b$ enforces s^r to be well aligned to edges extracted by EDN and prevents the foreground saliency pixels from being propagated to backgrounds.

5.3.2 Scribble Boosting

While we generate scribbles for a specific image, we simply annotate a very small portion of the foreground and background as shown in Fig. 5.1. Intra-class discontinuity, such as complex shapes and appearances of objects, may lead our model to be trapped in a local minima, with incomplete salient object segmented. Here, we attempt to propagate the scribble annotations to a denser annotation based on our initial estimation.

A straightforward solution to obtain denser annotations is to expand scribble labels by using DenseCRF [29], as shown in Fig. 5.7(c). However, as our scribble annotations are very sparse, DenseCRF fails to generate denser annotation from our scribbles (see Fig. 5.7(c)). As seen in Fig. 5.7(e), the predicted saliency map trained on (c) is still very similar to the one supervised by original scribbles (see Fig. 5.7(d)).

Instead of expanding the scribble annotation directly, we apply DenseCRF to our initial saliency prediction s^{init}, and update s^{init} to s^{crf}. Directly training a network with s^{crf} will introduce noise to the network as s^{crf} is not the exact ground-truth. We compute difference of s^{init} and s^{crf}, and define pixels with $s^{\text{init}} = s^{\text{crf}} = 1$ as foreground pixels in the new scribble annotation, $s^{\text{init}} = s^{\text{crf}} = 0$ as background pixels, and others as unknown pixels. In Fig. 5.7 (g) and Fig. 5.7 (h), we illustrate the intermediate results of scribble boosting. Note that, our method achieves better saliency prediction results than the case of applying DenseCRF to the initial prediction (see Fig. 5.7 (f)). This demonstrates the effectiveness of our scribble boosting scheme. In our experiments, after conducting one iteration of our scribble boosting step, our performance is almost on par with fully-supervised methods.

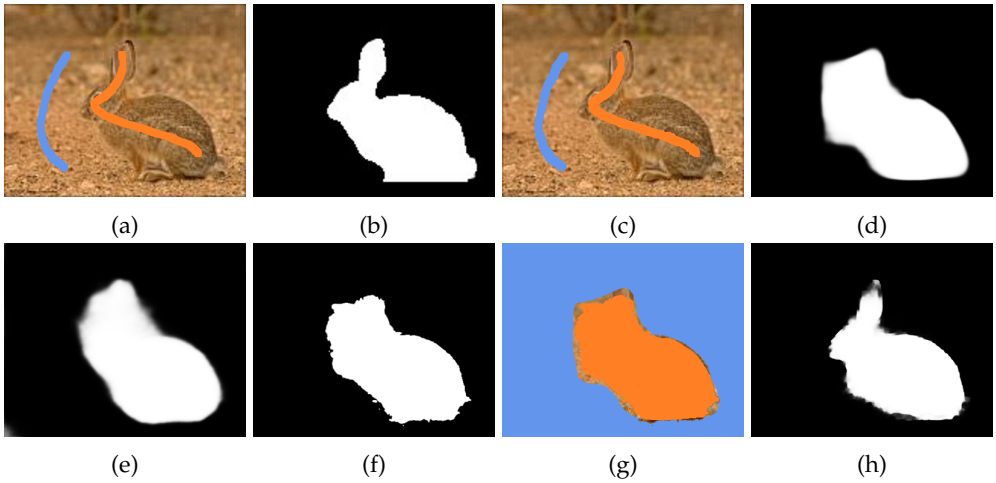


Figure 5.7: Illustration of using different strategies to enrich scribble annotations. (a) Input RGB image and scribble annotations. (b) Per-pixel wise ground-truth. (c) Result of applying DenseCRF to scribbles. (d) Saliency detection, trained on scribbles of (a). (e) Saliency detection, trained on scribbles of (c). (f) Applying DenseCRF to the result (d). (g) The confidence map between (d) and (f) for scribble boosting. Orange indicates consistent foreground, blue represents consistent background, and others are marked as unknown. (h) Our final result trained on new scribble (g).

5.3.3 Saliency Structure Measure

Existing saliency evaluation metrics (Mean Abosolute Error, Precision-recall curves, F-measure, E-measure [173] and S-measure [172]) focus on measuring accuracy of the prediction, while neglect whether a predicted saliency map complies with human perception or not. In other words, the estimated saliency map should be aligned with object structure of the input image. In [32], bIOU loss was proposed to penalize on saliency boundary length. We adapt the bIOU loss as an error metric B_μ to evaluate the structure alignment between saliency maps and their ground-truth.

Given a predicted saliency map s, and its pixel-wise ground truth y, their binarized edge maps are defined as g_s and g_y respectively. Then B_μ is expressed as: $B_\mu = 1 - \frac{2 \cdot \sum(g_s \cdot g_y)}{\sum(g_s^2 + g_y^2)}$, where $B_\mu \in [0, 1]$. $B_\mu = 0$ represents perfect prediction. As edges of prediction and ground-truth saliency maps may not be aligned well due to the small scales of edges, they will lead to unstable measurements (see Fig. 5.8). We dilate both edge maps with square kernel of size 3 before we compute the B_μ measure. As shown in Fig. 5.3, B_μ reflects the sharpness of predictions which is consistent with human perception.

5.3.4 Network Details

We use VGG16-Net [67] as our backbone network. In the edge detection network, we encode s_m to feature maps of channel size 32 through 1×1 convolutional layers. In the "DenseASPP" module (Fig. 5.5), the first three convolutional layers produce

Figure 5.8: The first two images show the original image edges. We dilate the original edges (last two images) to avoid misalignments due to the small scales of original edges.

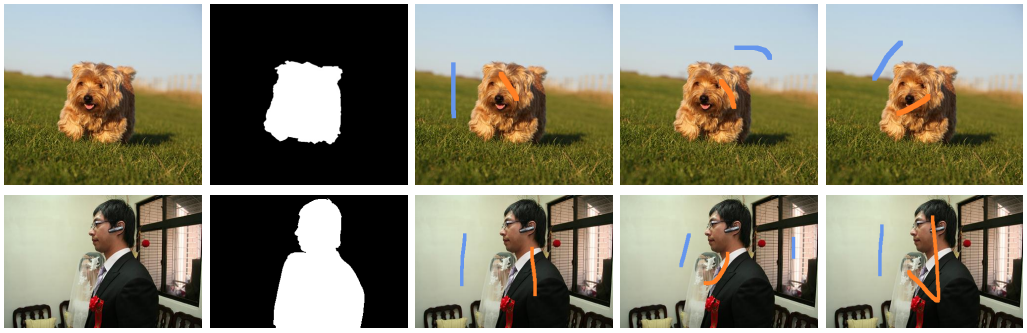


Figure 5.9: Illustration of scribble annotations by different labelers. From left to right: input RGB images, pixel-wise ground-truth labels, scribble annotations by three different labelers.

saliency features of channel size 32, and the last convolutional layer map the feature maps to s_5' of same size as s_5. Then we use two sequential convolutional layers to map s_5' to one channel coarse saliency map s^c. The hyper-parameters in Eq. 5.3 and Eq. 5.4 are set as: $\alpha = 10$, $\beta_1 = \beta_2 = 0.3$, $\beta_3 = 1$.

We train our model for 50 epochs using Pytorch, with the SPN initialized with parameters from VGG16-Net [67] pretrained on ImageNet [222]. The other newly added convolutional layers are randomly initialized with $\mathcal{N}(0, 0.01)$. The base learning rate is initialized as 1e-4. The whole training takes 6 hours with a training batch size 15 on a PC with a NVIDIA GeForce RTX 2080 GPU.

5.4 Experimental Results

5.4.1 Scribble Dataset

In order to train our weakly-supervised salient object detection method, we relabel an existing saliency dataset with scribble annotations by three annotators (S-DUTS dataset). In Fig. 5.9, we show two examples of scribble annotations by different labelers. Due to the sparsity of scribbles, the annotated scribbles do not have large overlaps. Thus, majority voting is not conducted. As aforementioned, labeling one image with scribbles is very fast, which only takes 1∼2 seconds on average.

Table 5.1: Evaluation results on six benchmark datasets. ↑ & ↓ denote larger and smaller is better, respectively.

		Fully Sup. Models											Weakly Sup./Unsup. Models					
	Metric	DGRL [181]	UCF [226]	PiCANet [31]	R3Net [227]	NLDF [32]	MSNet [39]	CPD [37]	AFNet [40]	PFAN [228]	PAGRN [38]	BASNet [33]	SBF [124]	WSI [98]	WSS [41]	MNL [66]	MSW [102]	**Ours**
ECSSD	B_μ ↓	.500	.699	.592	.472	.594	.542	.434	.510	.660	.574	**.364**	.759	.801	.808	.681	.851	**.550**
	F_β ↑	.903	.845	.872	**.914**	.871	.886	.908	.901	.859	.872	.913	.782	.762	.767	.810	.761	**.865**
	E_ξ ↑	.937	.887	.909	**.940**	.895	.922	.932	.929	.864	.887	.938	.835	.792	.796	.836	.788	**.908**
	$\mathcal{M}$ ↓	.043	.071	.054	.042	.066	.048	.043	.045	.047	.064	**.040**	.096	.068	.108	.090	.098	**.061**
DUT	B_μ ↓	.619	.812	.685	.606	.715	.642	.549	.603	.644	.645	**.480**	.812	.839	.830	.776	.890	**.655**
	F_β ↑	.726	.632	.711	.747	.683	.710	.739	.743	.701	.675	**.767**	.612	.641	.590	.597	.597	**.702**
	E_ξ ↑	.845	.760	.823	.853	.798	.831	.845	.846	.799	.772	**.865**	.763	.761	.729	.712	.728	**.835**
	$\mathcal{M}$ ↓	.063	.120	.072	.063	.080	.064	.057	.057	.062	.071	**.057**	.108	.100	.110	.103	.109	**.068**
PASCAL-S	B_μ ↓	.648	.783	.704	.662	.731	.671	.616	.659	.710	.692	**.582**	.815	.855	.831	.776	.870	**.665**
	F_β ↑	**.829**	.787	.799	.797	.793	.813	.822	.824	.754	.766	.821	.735	.653	.698	.748	.685	**.788**
	E_ξ ↑	**.835**	.795	.805	.781	.783	.822	.820	.827	.746	.755	.821	.746	.647	.690	.741	.693	**.798**
	$\mathcal{M}$ ↓	**.115**	.140	.128	.145	.145	.119	.122	.116	.137	.152	.122	.167	.206	.184	.158	.178	**.140**
HKU-IS	B_μ ↓	.496	.679	.561	.477	.553	.498	.421	.483	.530	.533	**.359**	.734	.782	.752	.627	.830	**.537**
	F_β ↑	.884	.819	.854	.892	.871	.878	.895	.888	.872	.864	**.903**	.783	.763	.773	.820	.734	**.858**
	E_ξ ↑	.939	.886	.910	.939	.914	.930	.940	.934	.898	.898	**.943**	.855	.800	.819	.858	.786	**.923**
	$\mathcal{M}$ ↓	.037	.062	.046	.036	.048	.039	.033	.036	.042	.048	**.032**	.075	.089	.079	.065	.084	**.047**
THUR	B_μ ↓	.578	-	.659	-	.652	.620	.524	.574	.743	.631	**.489**	.785	-	.788	.717	-	**.596**
	F_β ↑	.727	.	.710	-	.711	.718	**.750**	.733	.683	.740	.737	.627	-	.653	.691	-	**.718**
	E_ξ ↑	.838	.	.821	-	.827	.829	**.851**	.840	.804	.842	.841	.770	-	.775	.807	-	**.837**
	$\mathcal{M}$ ↓	.077	.	.084	-	.081	.079	.094	.072	.094	**.070**	.073	.107	-	.097	.086	-	**.0772**
DUTS	B_μ ↓	.564	.796	.635	-	.649	.582	.462	.540	.617	.587	**.400**	.808	.879	.780	.712	.829	**.603**
	F_β ↑	.790	.663	.757	-	.757	.792	**.825**	.812	.765	.778	.823	.622	.569	.633	.725	.648	**.747**
	E_ξ ↑	.887	.775	.853	-	.851	.883	**.902**	.893	.830	.842	.896	.763	.690	.806	.853	.742	**.865**
	$\mathcal{M}$ ↓	.051	.112	.062	-	.065	.049	**.043**	.046	.061	.056	.048	.107	.116	.100	.075	.091	**.062**

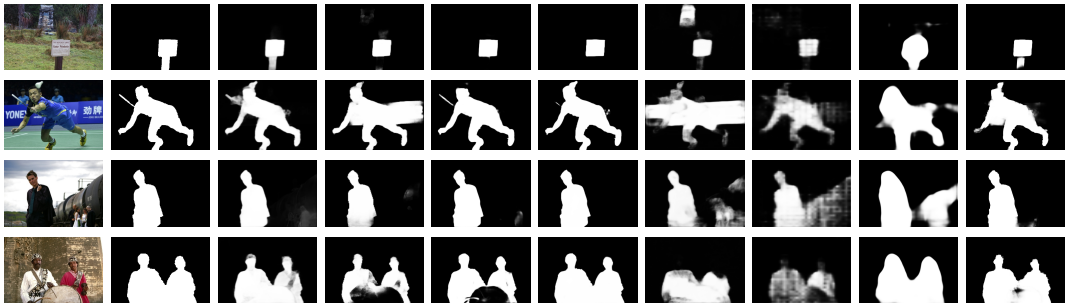


Figure 5.10: Comparisons of saliency maps. "M1" represents the results of a baseline model marked as "M1" in Section 5.4.4.

5.4.2 Setup

Datasets: We train our network on our newly labeled scribble saliency dataset: S-DUTS. Then, we evaluate our method on six widely-used benchmarks: (1) DUTS testing dataset [41]; (2) ECSSD [189]; (3) DUT [26]; (4) PASCAL-S [229]; (5) HKU-IS [30] and (6) THUR [190].

Competing methods: We compare our method with five state-of-the-art weakly-supervised/unsupervised methods and eleven fully-supervised saliency detection methods.

Evaluation Metrics: Four evaluation metrics are used, including Mean Absolute Error (MAE $\mathcal{M}$), Mean F-measure (F_β), mean E-measure (E_ξ [173]) and our proposed saliency structure measure (B_μ).

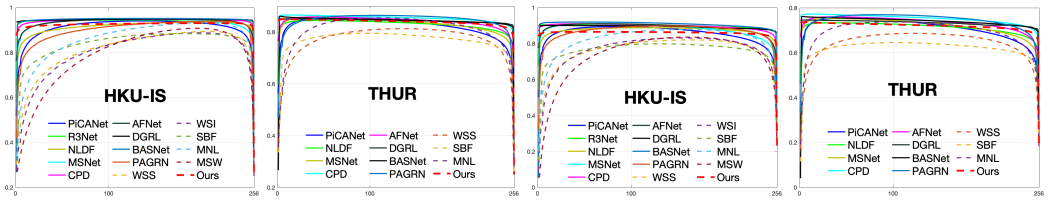


Figure 5.11: E-measure (1*st* two figures) and F-measure (last two figures) curves on two benchmark datasets. Best Viewed on screen.

5.4.3 Comparison with the State-of-the-Art

Quantitative Comparison: In Table 5.1 and Fig. 5.11, we compare our results with other competing methods. As indicated in Table 5.1, we achieves consistently the best performance compared with other weakly-supervised or unsupervised methods under these four saliency evaluation metrics. Since state-of-the-art weakly-supervised or unsupervised models do not impose any constraints on the boundaries of predicted saliency maps, these methods cannot preserve the structure in the prediction and produce high values on B_μ measure. In contrast, our method explicitly enforces a gated structure-aware loss to the edges of the prediction, and achieves lower B_μ. Moreover, our performance is also comparable or superior to some fully-supervised saliency models, such as DGRL and PiCANet. Fig. 5.11 shows the E-measure and F-measure curves of our method as well as the other competing methods on HKU-IS and THUR datasets. Due to limits of space, E-measure and F-measure curves on the other four testing datasets are provided in the supplementary material. As illustrated in Fig. 5.11, our method significantly outperforms the other weakly-supervised and unsupervised models with different thresholds, demonstrating the robustness of our method. Furthermore, the performance of our method is also on par with some fully-supervised methods as seen in Fig. 5.11.

Qualitative Comparison: We sample four images from the ECSSD dataset [189] and the saliency maps predicted by six competing methods and our method are illustrated in Fig. 5.10. Our method, while achieving performance on par with some fully-supervised methods, significantly outperforms other weakly-supervised and unsupervised models. In Fig. 5.10, we further show that directly training with scribbles produces saliency maps with poor localization ("M1"). Benefiting from our EDN as well as gated structure-aware loss, our network is able to produce sharper saliency maps than other weakly-supervised and unsupervised ones.

Table 5.2: Ablation study on six benchmark datasets.

	Metric	M0	M1	M2	M3	M4	M5	M6	M7	M8	M9
ECSSD	$B_\mu \downarrow$	.550	.896	.592	.616	.714	.582	.554	.771	.543	.592
	$F_\beta \uparrow$	.865	.699	.823	.804	.778	.845	.835	.696	.868	.839
	$E_\xi \uparrow$	.908	.814	.874	.859	.865	.898	.890	.730	.908	.907
	$\mathcal{M} \downarrow$	.061	.117	.083	.094	.091	.068	.074	.136	.059	. 070
DUT	$B_\mu \downarrow$	.655	.925	.696	.711	.777	.685	.665	.786	.656	.708
	$F_\beta \uparrow$	.702	.518	.656	.626	.580	.679	.658	.556	.691	.671
	$E_\xi \uparrow$	.835	.699	.807	.774	.743	.823	.805	.711	.823	.816
	$\mathcal{M} \downarrow$	.068	.134	.083	.102	.116	.074	.081	.108	.069	.080
PASCAL-S	$B_\mu \downarrow$	.665	.921	.732	.760	.787	.693	.676	.792	.664	.722
	$F_\beta \uparrow$	.788	.693	.748	.727	.741	.772	.768	.657	.792	.771
	$E_\xi \uparrow$	.798	.761	.757	.731	.795	.791	.782	.664	.800	.804
	$\mathcal{M} \downarrow$	.140	.171	.160	.173	.152	.145	.152	.204	.136	.143
HKU-IS	$B_\mu \downarrow$	.537	.892	.567	.609	.670	.574	.559	.747	.535	.564
	$F_\beta \uparrow$	.858	.651	.813	.789	.747	.835	.812	.646	.857	.821
	$E_\xi \uparrow$	.923	.799	.904	.878	.867	.911	.900	.761	.920	.907
	$\mathcal{M} \downarrow$	.047	.113	.060	.083	.080	.055	.062	.123	.047	.058
THUR	$B_\mu \downarrow$	.596	.927	.637	.677	.751	.635	.606	.780	.592	.650
	$F_\beta \uparrow$	.718	.520	.660	.641	.596	.696	.683	.586	.718	.690
	$E_\xi \uparrow$	.837	.687	.803	.773	.750	.824	.814	.718	.834	.804
	$\mathcal{M} \downarrow$	.077	.150	.099	.118	.123	.085	.087	.125	.078	.086
DUTS	$B_\mu \downarrow$	.603	.923	.681	.708	.763	.639	.634	.745	.604	.687
	$F_\beta \uparrow$	.747	.517	.688	.652	.607	.728	.685	.578	.743	.728
	$E_\xi \uparrow$	.865	.699	.833	.805	.776	.857	.828	.719	.856	.855
	$\mathcal{M} \downarrow$	.062	.135	.079	.101	.106	.068	.080	.106	.061	.080

5.4.4 Ablation Study

We carry out nine experiments (as shown in Table 5.2) to analyze our method, including our loss functions ("M1", "M2" and "M3"), network structure ("M4"), DenseCRF post-processing ("M5"), scribble boosting strategy ("M6"), scribble enlargement ("M7") and robustness analysis ("M8", "M9"). Our final result is denoted as "M0".

Direct training with scribble annotations: We employ the partial cross-entropy loss to train our SPN in Fig. 5.4 with scribble labels. The performance is marked as "M1". As expected, "M1" is much worse than our result "M0" and the high B_μ measure also indicates that object structure is not well preserved if only using the partial cross-entropy loss.

Impact of gated structure-aware loss: We add our gated structure-aware loss to "M1", and the performance is denoted by "M2". The gated structure-aware loss improves the performance in comparison with "M1". However, without using our EDN, "M2" is still inferior to "M0".

Impact of gate: We propose gated structure-aware loss to let the network focus on salient regions of images instead of the entire image as in the traditional smoothness loss [109]. To verify the importance of the gate, we compare our loss with the smoothness loss, marked as "M3". As indicated, "M2" achieves better performance than "M3", demonstrating the gate reduces the ambiguity of structure recovery.

Impact of the edge detection task: We add edge detection task to "M1", and use cross-entropy loss to train the EDN. Performance is indicated by "M4". We observe

that the B_μ measure is significantly decreased compared to "M1". This indicates that our auxiliary edge-detection network provides rich structure guidance for saliency prediction. Note that, our gated structure-aware loss is not used in "M4".

Impact of scribble boosting: We employ all the branches as well as our proposed losses to train our network and the performance is denoted by "M5". The predicted saliency map is also called our initial estimated saliency map. We observe decreased performance compared with "M0", where one iteration of scribble boosting is employed, which indicates effectiveness of the proposed boosting scheme.

Employing DenseCRF as post-processing: After obtaining our initial predicted saliency map, we can also use post-processing techniques to enhance the boundaries of the saliency maps. Therefore, we refine "M5" with DenseCRF, and results are shown in "M6", which is inferior to "M5". The reason lies in two parts: 1) the hyperparameters for DenseCRF is not the best; 2) DenseCRF recover structure information without considering saliency of the structure, causing extra false positive region. Using our scribble boosting mechanism, we can always achieve boosted or at least comparable performance as indicated by "M0".

Using Grabcut to generate pseudo label: Given scribble annotation, one can enlarge the annotation by using Grabcut [230]. We carried out experiment with pseudo label y' obtained by applying Grabcut to our scribble annotations y, and show performance in "M7". During training, we employ the same loss function as in Eq. 5.4, except that we use cross-entropy loss for $\mathcal{L}_s$. Performance of "M7" is worse than ours. The main reason is that pseudo label y' contains noise due to limited accuracy of Grabcut. Training directly with y' will overwhelm the network remembering the noisy label instead of learning useful saliency information.

Robustness to different scribble annotations: We report our performance "M0" by training the network with one set of scribble dataset. We then train with another set of the scribble dataset ("M8") to test robustness of our model. We observe staple performance compared with "M0". This implies that our method is robust to the scribble annotations despite their sparsity and few overlaps annotated by different labelers. We also conduct experiments with merged scribbles of different labelers as supervision signal and show performance of this experiment in the supplementary material.

Different edge detection methods: We obtain the edge maps E in Eq. 5.2 from RCF edge detection network [225] to train EDN. We also employ a hand-crafted edge map detection method, "Sobel", to train EDN, denoted by "M9". Since Sobel operator is more sensitive to image noise compared to RCF, "M9" is a little inferior to "M0". However, "M9" still achieves better performance than the results without using EDN, such as "M1", "M2" and "M3", which further indicates effectiveness of the edge detection module.

5.5 Conclusions

We proposed a weakly-supervised salient object detection (SOD) network trained on our newly labeled scribble dataset (S-DUTS). Our method significantly relaxes the

requirement of labeled data for training a SOD network. By introducing an auxiliary edge detection task and a gated structure-aware loss, our method produces saliency maps with rich structure, which is more consistent with human perception measured by our proposed saliency structure measure. Moreover, we develop a scribble boosting mechanism to further enrich scribble labels. Extensive experiments demonstrate that our method significantly outperforms state-of-the-art weakly-supervised or unsupervised methods and is on par with fully-supervised methods. Different from existing scribble-based dense prediction techniques that either explore only the boundary information [221] or mainly focus on modeling the image inherent pairwise term [220], we introduce an alternative solution, namely the structure-aware loss function, which serves as a self-supervised regularization. As an extension of our framework, we aim to achieve self-supervised learning based semantic segmentation via scribble annotation. Our main focus will be exploring the self-supervised learning techniques for effective structure modeling with the scribble annotations.

Chapter 6

Learning Noise-Aware Encoder-Decoder from Noisy Labels by Alternating Back-Propagation for Saliency Detection

To further relieve the labelling effort as in Chapter 5, we propose a noise-aware encoder-decoder framework to disentangle a clean saliency predictor from noisy training examples, where the noisy labels are generated by unsupervised handcrafted feature-based methods. Compared with the scribble annotations in Chapter 5, the noisy labels are automatically computed, with no human intervention, leading to a much cheaper setting. The proposed model consists of two sub-models parameterized by neural networks: (1) a saliency predictor that maps input images to clean saliency maps, and (2) a noise generator, which is a latent variable model that produces noises from Gaussian latent vectors. The whole model that represents noisy labels is a sum of the two sub-models. The goal of training the model is to estimate the parameters of both sub-models, and simultaneously infer the corresponding latent vector of each noisy label. We propose to train the model by using an alternating back-propagation (ABP) algorithm, which alternates the following two steps: (1) learning back-propagation for estimating the parameters of two sub-models by gradient ascent, and (2) inferential back-propagation for inferring the latent vectors of training noisy examples by Langevin Dynamics. To prevent the network from converging to trivial solutions, we utilize an edge-aware smoothness loss to regularize hidden saliency maps to have similar structures as their corresponding images. Experimental results on several benchmark datasets indicate the effectiveness of the proposed model.

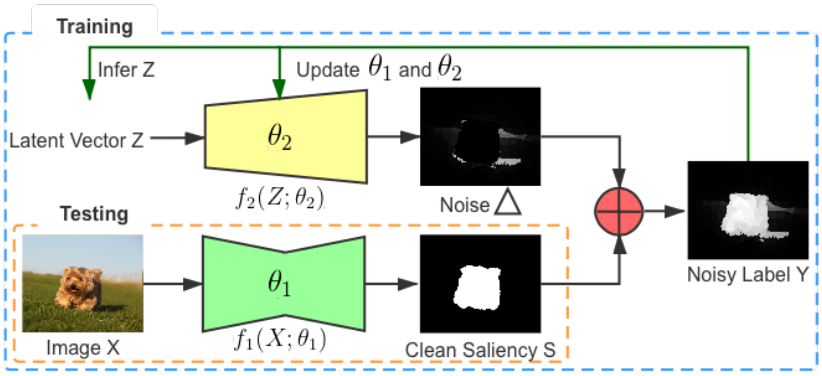


Figure 6.1: An illustration of our framework. Representation: Each noisy label Y is represented as a sum of a clean saliency S and a noise map Δ. The clean saliency S is predicted from an image X by an encoder-decoder network f_1, and the noise is produced from a Gaussian noise vector Z by a generator network f_2. Training: given the observed image X and the corresponding noisy label Y, (i) the latent vector Z is inferred by MCMC and (ii) the parameters $\{\theta_1, \theta_2\}$ of the encoder-decoder f_1 and the generator f_2 are updated by the gradient ascent for maximum likelihood. Testing: once the model is learned, the disentangled salicey predictor f_1 is the desired model for salicey prediction.

6.1 Introduction

Visual saliency detection aims to locate salient regions that attract human attention. Conventional saliency detection methods [24, 26] rely on human designed features to compute saliency for each pixel or superpixel. The deep learning revolution makes it possible to train end-to-end deep saliency detection models in a data-driven manner [31, 36, 37, 38, 39, 40, 33, 231, 34, 129, 35, 212, 232, 60], outperforming handcrafted feature-based solutions by a wide margin. However, the success of deep models mainly depends on a large amount of accurate human labeling [41, 42, 233], which is typically expensive and time-consuming.

To relieve the burden of pixel-wise labeling, weakly supervised [98, 41, 63] and unsupervised saliency detection models [66, 124, 99] have been proposed. The former direction focuses on learning saliency from cheap but clean annotations, while the latter one studies learning saliency from noisy labels, which are typically obtained by conventional handcrafted feature-based methods. We follow the second direction and propose a deep latent variable model that we call the noise-aware encoder-decoder to disentangle a clean saliency predictor from noisy labels. In general, a noisy label can be (1) a coarse saliency label generated by algorithmic pipelines using handcrafted features, (2) an imperfect human-annotated saliency label, or even (3) a clean label, which actually is a special case of noisy label, in which noise is none. Aiming at unsupervised saliency prediction, our paper assumes noisy labels

to be produced by unsupervised handcrafted feature-based saliency methods, and places emphasis on disentangled representation of noisy labels by the noise-aware encoder-decoder.

Given a noisy dataset $D = \{(X_i, Y_i)\}_{i=1}^n$ of n examples, where X and Y are image and its corresponding noisy saliency label, we intend to disentangle noise Δ and clean saliency S from each noisy label Y, and learn a clean saliency predictor $f_1 : X \to S$. To achieve this, we propose a conditional latent variable model, which is a disentangled representation of noisy saliency Y. See Figure 6.1 for an illustration of the proposed model. In the context of the model, each noisy label is assumed to be generated by adding a specific noise or perturbation Δ to its clean saliency map S that is dependent on its image X. Specifically, the model consists of two sub-models: (1) saliency predictor f_1: an encoder-decoder network that maps an input image X to a latent clean saliency map S, and (2) noise generator f_2: a top-down neural network that produces a noise or error Δ from a low-dimensional Gaussian latent vector Z.

As a latent variable model, the rigorous maximum likelihood learning (MLE) typically requires to compute an intractable posterior distribution, which is an inference step. To learn the latent variable model, two algorithms can be adopted: variational auto-encoder (VAE) [50] or alternating back-propagation (ABP) [139], [234], [235]. VAE approximates MLE by minimizing the evidence lower bound with a separate inference model to approximate the true posterior, while ABP directly targets MLE and computes the posterior via Markov chain Monte Carlo (MCMC). We generalize the ABP algorithm to learn the proposed model, which alternates the following two steps: (1) learning back-propagation for estimating the parameters of two sub-models, and (2) inferential back-propagation for inferring the latent vectors of training examples. As there may exist infinite combinations of S and Δ such that $S + \Delta$ perfectly matches the provided noisy label Y, we further adopt the edge-aware smoothness loss [109] to serve as a regularization to force each latent saliency map S to have a similar structure as its input image X. The learned disentangled saliency predictor f_1 is the desired model for testing.

Our solution is different from existing weak or noisy label-based saliency approaches [66, 124, 99, 236] in the following aspects: Firstly, unlike [66], we don't assume the saliency noise distribution is a Gaussian distribution. Our noise generator parameterized by a neural network is flexible enough to approximate any forms of structural noises. Secondly, we design a trainable noise generator to explicitly represent each noise Δ as a non-linear transformation of low-dimensional Gaussian noise Z, which is a latent variable that need to be inferred during training, while [66, 124, 99, 236] have no noise inference process. Thirdly, we have no constraints on the number of noisy labels generated from each image, while [66, 124, 99] require multiple noisy labels per image for noise modeling or pseudo label generation. Lastly, our edge-aware smoothness loss serves as a regularization to force the produced latent saliency maps to be well aligned with their input images, which is different from [236], where object edges are used to produce pseudo saliency labels via multi-scale combinatorial grouping (MCG) [237].

Our main contributions can be summarized as follows:

- We propose to learn a clean saliency predictor from noisy labels by a novel latent variable model that we call noise-aware encoder-decoder, in which each noisy label is represented as a sum of the clean saliency generated from the input image and a noise map generated from a latent vector.
- We propose to train the proposed model by an alternating back-propagation (ABP) algorithm, which rigorously and efficiently maximizes the data likelihood without recruiting any other auxiliary model.
- We propose to use an edge-aware smoothness loss as a regularization to prevent the model from converging to a trivial solution.
- Experimental results on various benchmark datasets show the state-of-the-art performances of our framework in the task of unsupervised saliency detection, and also comparable performances with the existing fully-supervised saliency detection methods.

6.2 Related Work

Fully supervised saliency detection models [231, 34, 129, 35, 212, 33, 37, 215, 131] mainly focus on designing networks that utilize image context information, multi-scale information, and image structure preservation. [231] introduces feature polishing modules to update each level of features by incorporating all higher levels of context information. [34] presents a cross feature module and a cascaded feedback decoder to effectively fuse different levels of features with a position-aware loss to penalize the boundary as well as pixel dissimilarity between saliency outputs and labels during training. [35] proposes a saliency detection model that integrates both top-down and bottom-up saliency inferences in an iterative and cooperative manner. [212] designs a pyramid attention structure with an edge detection module to perform edge-preserving salient object detection. [33] uses a hybrid loss for boundary-aware saliency detection.

Learning saliency models without pixel-wise labeling can relieve the burden of costly pixel-level labeling. Those methods train saliency detection models with low-cost labels, such as image-level labels [41, 98, 102], noisy labels [66, 124, 99], object contours [236], scribble annotations [63], *etc.*[41] introduces a foreground inference network to produce initial saliency maps with image-level labels, which are further refined and then treated as pseudo labels for iterative training. [124] fuses saliency maps from unsupervised handcrafted feature-based methods with heuristics within a deep learning framework. [66] collaboratively updates a saliency prediction module and a noise module to achieve learning saliency from multiple noisy labels. In [99], the initial noisy labels are refined by a self-supervised learning technique, and then treated as pseudo labels. [236] creates a contour-to-saliency network, where saliency masks are generated by its contour detection branch via MCG [237] and then those generated saliency masks are further used to train its saliency detection branch.

Learning from noisy labels techniques mainly focus on three main directions: (1) developing regularization [238, 79]; (2) estimating the noise distribution by assuming that noisy labels are corrupted from clean labels by an unknown noise transition matrix [114, 239] and (3) training on selected samples [118, 120]. [238] deals with noisy labeling by augmenting the prediction objective with a notion of perceptual consistency. [79] proposes a framework to solve noisy label problem by updating both model parameters and labels. [239] proposes to simultaneously learn the individual annotator model, which is represented by a confusion matrix, and the underlying true label distribution (i.e., , classifier) from noisy observations. [118] proposes to learn an extra network called MentorNet to generate a curriculum, which is a sample weighting scheme, for the base ConvNet called StudentNet. The generated curriculum helps the StudentNet to focus on those samples whose labels are likely to be correct.

6.3 Proposed Framework

The proposed model consists of two sub-models: (1) a saliency predictor, which is parameterized by an encoder-decoder network that maps the input image X to the clean saliency S; (2) a noise generator, which is parameterized by a top-down generator network that produces a noise or error Δ from a Gaussian latent vector Z. The resulting model is a sum of the two sub-models. Given training images with noisy labels, the MLE training of the model leads to an alternating back-propagation algorithm, which will be introduced in details in the following sections. The learned encoder-decoder network, which takes as input an image X and outputs its clean saliency S, is the disentangled model for saliency detection.

6.3.1 Noise-Aware Encoder-Decoder Network

Let $D = \{(X_i, Y_i)\}_{i=1}^n$ be the training dataset, where X is the training image, Y is the noisy label of X, n is the size of the training dataset. Formally, the noise-aware encoder-decoder model can be formulated as follows:

$$S = f_1(X; \theta_1), \tag{6.1}$$

$$\Delta = f_2(Z; \theta_2), Z \sim \mathcal{N}(0, I_d), \tag{6.2}$$

$$Y = S + \Delta + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2 I_D), \tag{6.3}$$

where f_1 in Eq. (6.1) is an encoder-decoder structure parameterized by θ_1 for saliency detection. It takes as input an image X and predicts its clean saliency map S. Eq. (6.2) defines a noise generator, where Z is a low-dimensional Gaussian noise vector following $\mathcal{N}(0, I_d)$ (I_d is the d-dimensional identity matrix) and f_2 is a top-down deconvolutional neural network parametrized by θ_2 that generates a saliency noise Δ from the noise vector Z. In Eq. (6.3), we assume that the observed noisy label Y is a sum of the clean saliency map S and the noise Δ, plus a Gaussian residual $\epsilon \sim \mathcal{N}(0, \sigma^2 I_D)$, where we assume σ is given and I_D is the D-dimensional identity

matrix. Although Z is a Gaussian noise, the generated noise Δ is not necessarily Gaussian due to the non-linear transformation f_2.

We call our network the noise-aware encoder-decoder network as it explicitly decomposes a noisy label Y into a noise Δ and a clean label S, and simultaneously learns a mapping from the image X to the clean saliency map S via an encoder-decoder network as shown in Fig. 6.1. Since the resulting model involves latent variables Z, training the model by maximum likelihood learning typically needs to learn the parameters θ_1 and θ_2, and also infer the noise latent variable Z_i for each observed data pair (X_i, Y_i). The noise and the saliency information are disentangled once the model is learned. The learned encoder-decoder sub-model $S = f_1(X; \theta_1)$ is the desired saliency detection network.

6.3.2 Maximum Likelihood via Alternating Back-Propagation

For notation simplicity, let $f = \{f_1, f_2\}$ and $\theta = \{\theta_1, \theta_2\}$. The proposed model is rewritten as a summarized form: $Y = f(X, Z; \theta) + \epsilon$, where $Z \sim \mathcal{N}(0, I_d)$ and ϵ is the observation error. Given a dataset $D = \{(X_i, Y_i)\}_{i=1}^n$, each training example (X_i, Y_i) should have a corresponding Z_i, but all data shares the same model parameter θ. Intuitively, we should infer Z_i and learn θ to minimize the reconstruction error $\sum_{i=1}^n \|Y_i - f(X_i, Z_i; \theta)\|^2$ based on our formulation in Section 6.3.1. More formally, the model seeks to maximize the observed-data log-likelihood: $\mathcal{L}(\theta) = \sum_{i=1}^n \log p_\theta(Y_i|X_i)$. Specifically, let $p(Z)$ be the prior distribution of Z. Let $p_\theta(Y|X, Z) \sim \mathcal{N}(f(X, Z; \theta), \sigma^2 I)$ be the conditional distribution of the noisy label Y given Z and X. The conditional distribution of Y given X is $p_\theta(Y|X) = \int p(Z) p_\theta(Y|X, Z) dZ$ with the latent variable Z integrated out.

The gradient of $\mathcal{L}(\theta)$ can be calculated according to the following identity:

$$\frac{\partial}{\partial \theta} \log p_\theta(Y|X) = \frac{1}{p_\theta(Y|X)} \frac{\partial}{\partial \theta} p_\theta(Y|X) = \mathrm{E}_{p_\theta(Z|Y,X)} \left[\frac{\partial}{\partial \theta} \log p_\theta(Y, Z|X) \right]. \tag{6.4}$$

The expectation term $\mathrm{E}_{p_\theta(Z|Y,X)}$ is analytically intractable. The conventional way of training such a latent variable model is the variational inference, in which the intractable posterior distribution $p_\theta(Z|Y, X)$ is approximated by an extra trainable tractable neural network $p_\phi(Z|Y, X)$. We resort to Monte Carlo average through drawing samples from the posterior distribution $p_\theta(Z|Y, X)$. This step corresponds to inferring the latent vector Z of the generator for each training example. Specifically, we use Langevin Dynamics [140] (a gradient-based Monte Carlo method) to sample Z. The Langevin Dynamics for sampling $Z \sim p_\theta(Z|Y, X)$ iterates:

$$Z_{t+1} = Z_t + \frac{s^2}{2} \left[\frac{\partial}{\partial Z} \log p_\theta(Y, Z_t|X) \right] + s\mathcal{N}(0, I_d), \tag{6.5}$$

with

$$\frac{\partial}{\partial Z} \log p_\theta(Y, Z|X) = \frac{1}{\sigma^2}(Y - f(X, Z; \theta)) \frac{\partial}{\partial Z} f(X, Z) - Z, \tag{6.6}$$

where t and s are the time step and step size of the Langevin Dynamics respectively. In each training iteration, for a given data pair (X_i, Y_i), we run l steps of Langevin Dynamics to infer Z_i. The Langevin Dynamics is initialized with Gaussian white noise (i.e., , cold start) or the result of Z_i obtained from the previous iteration (i.e., , warm start). With the inferred Z_i along with (X_i, Y_i), the gradient used to update the model parameters θ is:

$$\frac{\partial}{\partial \theta}\mathcal{L}(\theta) \approx \sum_{i=1}^{n} \frac{\partial}{\partial \theta} \log p_\theta(Y_i, X_i|Z_i), = \sum_{i=1}^{n} \frac{1}{\sigma^2}(Y_i - f(X_i, Z_i; \theta))\frac{\partial}{\partial \theta} f(X_i, Z_i). \tag{6.7}$$

Algorithm 4 Alternating back-propagation for noise-aware encoder-decoder

Input: Dataset with noisy labels $D = \{(X_i, Y_i)\}_{i=1}^n$, learning epochs K, number of Langevin steps l, Langevin step size s, learning rate γ

Output: Network parameters $\theta = \{\theta_1, \theta_2\}$, and the inferred latent vectors $\{Z_i\}_{i=1}^n$

1: Initialize θ_1 with the VGG16-Net[67] for image classification, θ_2 with a truncated Gaussian distribution, and Z_i with a standard Gaussian distribution.
2: **for** $k = 1, ..., K$ **do**
3: **Inferential back-propagation**: For each i, run l steps of Langevin Dynamics with a step size s to sample $Z_i \sim p_\theta(Z_i|Y_i, X_i)$ following Eq. (6.5), with Z_i initialized as a Gaussian white noise or the result from previous iteration.
4: **Learning back-propagation**: Update model parameters θ by Adam [240] optimizer with a learning rate γ and the gradient $\frac{\partial}{\partial \theta}[\mathcal{L}(\theta) - \lambda l_s(X, S; \theta)]$, where the gradient of $\mathcal{L}(\theta)$ is computed according to Eq. (6.7).
5: **end for**

To encourage the latent output S of the encoder-decoder f_1 to be a meaningful saliency map, we add a negative edge-aware smoothness loss [109] defined on S to the log-likelihood objective $\mathcal{L}(\theta)$. The smoothness loss serves as a regularization term to avoid a trivial decomposition of S and Δ given Y. Following [109], we use first-order derivatives (i.e., , edge information) of both the latent clean saliency map S and the input image X to compute the smoothness loss

$$l_s(X, S) = \sum_{u,v} \sum_{d \in x,y} \Psi(|\partial_d S_{u,v}| e^{-\alpha|\partial_d X_{u,v}|}), \tag{6.8}$$

where Ψ is the Charbonnier penalty formula, defined as $\Psi(s) = \sqrt{s^2 + 1e^{-6}}$, (u, v) represent pixel coordinates, and d indexes over the partial derivative in x and y directions. We estimate θ by gradient ascent on $\mathcal{L}(\theta) - \lambda l_s(X, S; \theta)$. In practice, we set $\lambda = 0.7$, and $\alpha = 10$ in Eq. (6.8).

The whole process of updating both $\{Z_i\}$ and $\theta = \{\theta_1, \theta_2\}$ is summarized in Algorithm 4, which is implemented as alternating back-propagation, because both gradients in Eq. (6.5) and (6.7) can be computed via back-propagation.

6.3.3 Comparison with Variational Inference

The proposed model can also be learned in a variational inference framework, where the intractable $p_\theta(Z|Y,X)$ in Eq. 6.4 is approximated by a tractable $q_\phi(Z|Y,X)$, such as $q_\phi(Z|Y,X) \sim \mathcal{N}(\mu_\phi(Y,X), \text{diag}(v_\phi(Y,X)))$, where both μ_ϕ and v_ϕ are bottom-up networks that map (X,Y) to Z, with ϕ standing for all parameters of the bottom-up networks. The objective of variational inference is:

$$\begin{aligned} &\min_\theta \min_\phi \text{KL}(q_{\text{data}}(Y|X)p_\phi(Z|Y,X)\|p_\theta(Z,Y|X)) = \\ &\min_\theta \min_\phi \text{KL}(q_{\text{data}}(Y|X)\|p_\theta(Y|X)) + \text{KL}(p_\phi(Z|Y,X)\|p_\theta(Z|Y,X)). \end{aligned} \tag{6.9}$$

Recall that the maximum likelihood learning in our algorithm is equivalent to minimizing $\text{KL}(q_{\text{data}}(Y|X)\|p_\theta(Y|X))$, where $q_{\text{data}}(Y|X)$ is the conditional training data distribution. The accuracy of variational inference in Eq. 6.9 depends on the accuracy of an approximation of the true posterior distribution $p_\theta(Z|Y,X)$ by the inference model $p_\phi(Z|Y,X)$. Theoretically, the variational inference is equivalent to the maximum likelihood solution, when $\text{KL}(p_\phi(Z|Y,X)\|p_\theta(Z|Y,X)) = 0$. However, in practice, there is always a gap between them due to the design of the inference model and the optimization difficulty. Therefore, without relying on an extra assisting model, our alternating back-propagation algorithm is more natural, straightforward and computationally efficient than variational inference. We refer readers to [241] for a comprehensive tutorial on latent variable models.

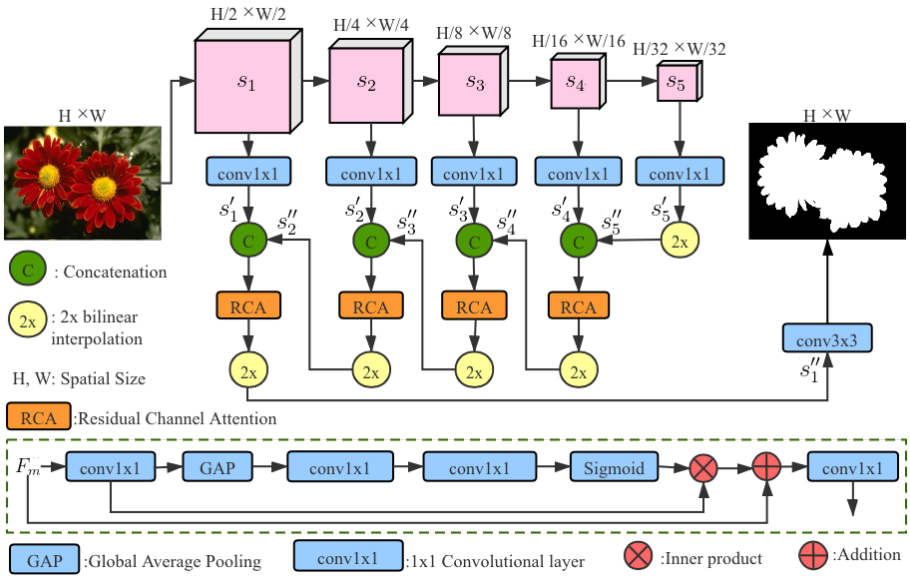


Figure 6.2: An illustration of the encoder-decoder-based saliency detection network (Green part in Fig.6.1).

6.3.4 Network Architectural Design

We now introduce the architectural designs of the encoder-decoder network (f_1 in Eq. 6.1, or the green encoder-decoder in Fig. 6.1) and the noise generator network (f_2 in Eq. 6.2, or the yellow decoder in Fig. 6.1) in this section.

Noise Generator: We construct the noise generator by using four cascaded deconvolutional layers, with a tanh activation function at the end to generate a noise map Δ in the range of $[-1, 1]$. Batch normalization and ReLU layers are added between two nearby deconvolutional layers. The dimensionality of the latent variable $d = 8$.

Encoder-Decoder Network: Most existing deep saliency prediction networks are based on widely used backbone networks, including the VGG16-Net [67], ResNet [28], etc. Due to stride operations and multiple pooling layers used in these deep architectures, the saliency maps that are generated directly using the above backbone networks are low in spatial resolution, causing blurred edges. To overcome this, we propose an encoder-decoder-based framework with the VGG16-Net [67] as the backbone as shown in Fig. 6.2. We denote the last convolutional layer of each convolutional group of VGG16-Net by $s_1, s_2, ..., s_5$ (corresponding to "relu1_2", "relu2_2", "relu3_3", "relu4_3", and "relu5_3", respectively). To reduce the channel dimension of s_m, a 1×1 convolutional layer is used to transform s_m to s'_m of channel dimension 32. Then a Residual Channel Attention (RCA) module [163] is adopted to effectively fuse the intermediate high- and low-level features. Specifically, given the high- and low-level feature maps s'_m and s'_{m-1}, we first upsample s'_m to s''_m, which has the same spatial resolution as s'_{m-1}, by bilinear interpolation. Then we concatenate s''_m and s'_{m-1} to form a new feature map F_m. Similar to [163], we feed F_m to the RCA block to achieve the discriminative feature extraction. Inside each channel attention block, we perform "squeeze and excitation" [242] by first "squeezing" the input feature map F_m to be half of the original channel size to obtain better nonlinear interactions across channels, and then "exciting" the squeezed feature map back to the original channel size. By adding a 3×3 convolutional layer to the lowest level of the RCA module, we obtain a one-channel saliency map $S_i = f_1(X_i; \theta_1)$.

6.4 Experiments

6.4.1 Experimental Setup

Datasets: We evaluate our performance on five saliency benchmark datasets. We use 10,553 images from the DUTS dataset [41] for training, and we generate noisy labels from images using handcrafted feature based-methods, such as RBD [24], MR [26] and GS [183] due to their high efficiencies. Testing datasets include the DUTS testing set, ECSSD [189], DUT [26], HKU-IS [30] and THUR [190].

Evaluation Metrics: Four metrics are used to evaluate the performance of our method and the competing methods, including two widely used metrics, i.e., , Mean Absolute Error ($\mathcal{M}$) and mean F-measure (F_β), and two newly released structure-

Table 6.1: Benchmarking performance comparison. Bold numbers represent best performance methods. ↑ & ↓ denote larger and smaller is better, respectively.

		Fully Suppervised Models							Weakly Sup./Unsup. Models					
	Metric	DGRL [181]	NLDF [32]	MSNet [39]	CPD [37]	AFNet [40]	SCRN [130]	BASNet [33]	C2S [236]	WSI [98]	WSS [41]	MNL [66]	MSW [102]	Ours
DUTS	$S_\alpha \uparrow$	.8460	.8162	.8617	.8668	.8671	**.8848**	.8657	.8049	.6966	.7484	.8128	.7588	**.8276**
	$F_\beta \uparrow$	.7898	.7567	.7917	.8246	.8123	**.8333**	.8226	.7182	.5687	.6330	.7249	.6479	**.7467**
	$E_\xi \uparrow$	.8873	.8511	.8829	**.9021**	.8928	.8996	.8955	.8446	.6900	.8061	.8525	.7419	**.8592**
	$\mathcal{M} \downarrow$	.0512	.0652	.0490	.0428	.0457	**.0398**	.0476	.0713	.1156	.1000	.0749	.0912	**.0601**
ECSSD	$S_\alpha \uparrow$	.9019	.8697	.9048	.9046	.9074	**.9204**	.9104	-	.8049	.8081	.8456	.8246	**.8603**
	$F_\beta \uparrow$	.8978	.8714	.8856	.9076	.9008	.9103	**.9128**	-	.7621	.7744	.8098	.7606	**.8519**
	$E_\xi \uparrow$	.9336	.8955	.9218	.9321	.9294	.9333	**.9378**	-	.7921	.8008	.8357	.7876	**.8834**
	$\mathcal{M} \downarrow$	.0447	.0655	.0479	.0434	.0450	.0407	**.0399**	-	.1137	.1055	.0902	.0980	**.0712**
DUT	$S_\alpha \uparrow$	.8097	.7704	.8093	.8177	.8263	**.8365**	.8362	.7731	.7591	.7303	.7332	.7558	**.7914**
	$F_\beta \uparrow$	.7264	.6825	.7095	.7385	.7425	.7491	**.7668**	.6649	.6408	.5895	.5966	.5970	**.7007**
	$E_\xi \uparrow$	.8446	.7983	.8306	.8450	.8456	.8474	**.8649**	.8100	.7605	.7292	.7124	.7283	**.8158**
	$\mathcal{M} \downarrow$	.0632	.0796	.0636	.0567	.0574	**.0560**	.0565	.0818	.0999	.1102	.1028	.1087	**.0703**
HKU-IS	$S_\alpha \uparrow$	.8968	.8787	.9065	.9039	.9053	**.9158**	.9089	.8690	.8079	.8223	.8602	.8182	**.8901**
	$F_\beta \uparrow$	.8844	.8711	.8780	.8948	.8877	.8942	**.9025**	.8365	.7625	.7734	.8196	.7337	**.8782**
	$E_\xi \uparrow$	.9388	.9139	.9304	.9402	.9344	.9351	**.9432**	.9103	.7995	.8185	.8579	.7862	**.9191**
	$\mathcal{M} \downarrow$	.0374	.0477	.0387	.0333	.0358	.0337	**.0322**	.0527	.0885	.0787	.0650	.0843	**.0428**
THUR	$S_\alpha \uparrow$	.8162	.8008	.8188	.8311	.8251	**.8445**	.8232	.7922	-	.7751	.8041	-	**.8101**
	$F_\beta \uparrow$	.7271	.7111	.7177	.7498	.7327	**.7584**	.7366	.6834	-	.6526	6911	-	**.7187**
	$E_\xi \uparrow$	.8378	.8266	.8288	.8514	.8398	**.8575**	.8408	.8107	-	.7747	.8073	-	**.8378**
	$\mathcal{M} \downarrow$	.0774	.0805	.0794	**.0635**	.0724	.0663	.0734	.0890	-	.0966	.0860	-	**.0703**

aware metrics: mean E-measure (E_ξ) [173] and S-measure (S_α) [172].

Training Details: Each input image is rescaled to 352×352 pixels. The encoder part in Fig. 6.2 is initialized using the VGG16-Net weights pretrained for image classification [67]. The weights of other layers are initialized using the "truncated Gaussian" policy, and the biases are initialized to be zeros. We use the Adam [240] optimizer with a momentum equal to 0.9, and decrease the learning rate γ by 10% after running 80% of the maximum epochs $K = 20$. The learning rate is initialized to be 0.0001. The number of Langevin steps l is 6. The Langevin step size s is 0.3. The σ in Eq.(6.3) is 0.1. The whole training takes 8 hours with a batch size 10 on a PC with an NVIDIA GeForce RTX GPU. We use the PaddlePaddle [243] deep learning platform.

6.4.2 Comparison with the State-of-the-art Methods

We compare our method with seven fully supervised deep saliency prediction models and five weakly supervised/unsupervised saliency prediction models, and their performances are shown in Table 6.1 and Fig. 6.3. Table 6.1 shows that compared with the weakly supervised/unsupervised models, the proposed method achieves the best performance, especially on DUTS and HKU-IS datasets, where our method achieves an approximately 2% performance improvement for S-measure, and a 4% improvement for mean F-measure. Further, the proposed method even achieves com-

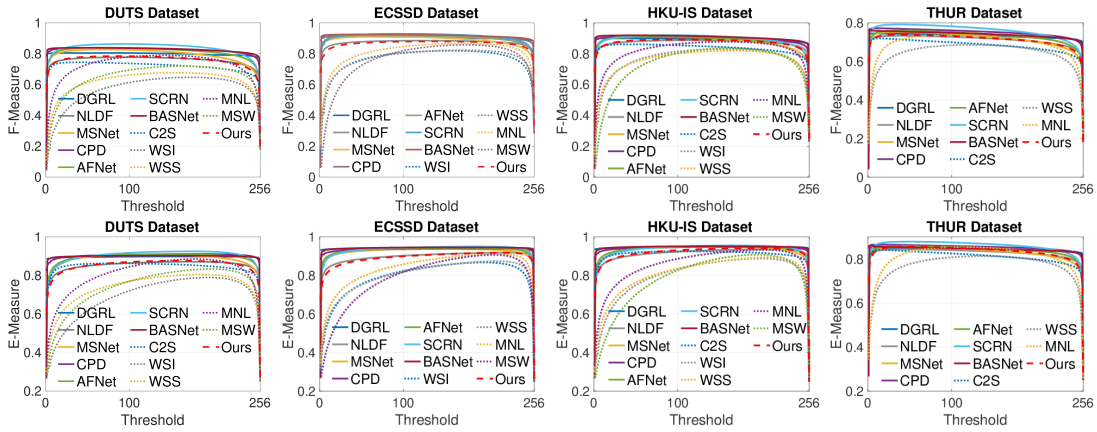


Figure 6.3: F-measure and E-measure curves on four datasets (DUTS, ECSSD, HKU-IS, THUR). Best viewed in color on screen.

parable performances with some newly released fully supervised models. For example, we achieve comparable performance with NLDF [32] and DGRL [181] on all the five benchmark datasets. Fig.6.3 shows the 256-dimensional F-measure and E-measure (where the x-axis represents threshold for saliency map binarization) of our method and the competing methods on four datasets, where the weakly supervised/unsupervised methods are represented by dotted curves. We can observe that the performances of the fully supervised models are better than those of the weakly supervised/unsupervised models. As shown in Fig.6.3, our performance shows stability with different thresholds relative to the existing methods, indicating the robustness of our model.

Figure 6.4 demonstrates a qualitative comparison on several challenging cases. For example, the salient object in the first row is large, and connects to the image border. Most competing methods fail to segment the border-connected region, while our method almost finds the whole salient region in this case. Also, salient object in the second row has a long and narrow shape, which is challenging to some competing methods. Our method performs very well and precisely detect the salient object.

6.4.3 Ablation Study

We conduct the following experiments for an ablation study.

(1) Encoder-decoder f_1 only: To study the effect of the noise generator, we evaluate the performance of the encoder-decoder (as shown in Fig. 6.2) directly learned from the noisy labels, without noise modeling or smoothness loss. The performance is shown in Table 6.2 with a label "f_1", which is clearly worse than ours. This result is also consistent with the conclusion that deep neural networks is not robust to noise [244].

(2) Encoder-decoder f_1 + smoothness loss l_s: As an extension of method "f_1", one can add the smoothness loss in Eq. (6.8) as a regularization to better use the

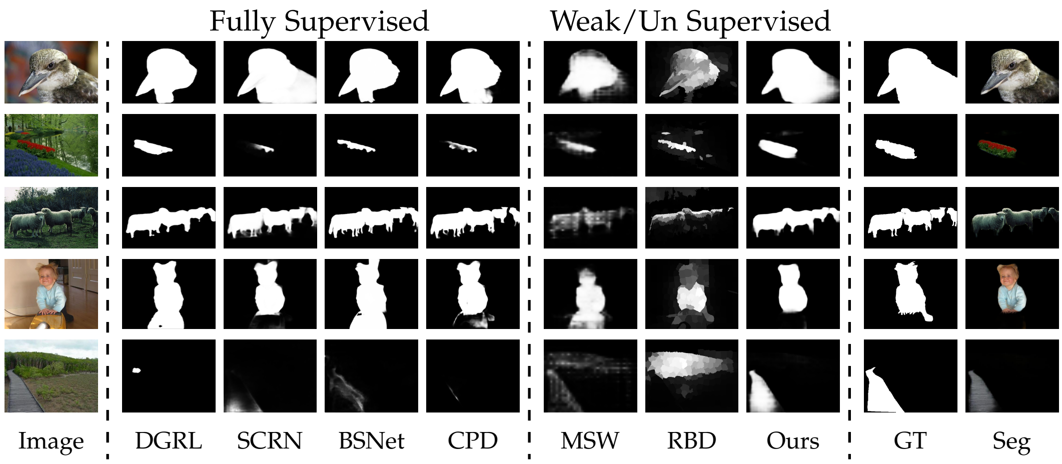


Figure 6.4: Comparison of saliency predictions, where each row displays an input image, its predicted saliency maps by four fully supervised competing methods (DGRL, SCRN, BASNet, and CPD), one weakly (MSW) and one unsupervised (RBD) methods, our prediction (Ours), the ground truth (GT) saliency map and our segmented foreground image (Seg).

Table 6.2: Ablation study. Some certain key components of the model are removed and the learned model is evaluated for saliency prediction in terms of S_α, F_β, E_ξ, and $\mathcal{M}$. $\uparrow$ & $\downarrow$ denote larger and smaller is better, respectively.

	DUTS				ECSSD				DUT				HKU-IS				THUR			
Model	$S_\alpha\uparrow$	$F_\beta\uparrow$	$E_\xi\uparrow$	$\mathcal{M}\downarrow$	$S_\alpha\uparrow$	$F_\beta\uparrow$	$E_\xi\uparrow$	$\mathcal{M}\downarrow$	$S_\alpha\uparrow$	$F_\beta\uparrow$	$E_\xi\uparrow$	$\mathcal{M}\downarrow$	$S_\alpha\uparrow$	$F_\beta\uparrow$	$E_\xi\uparrow$	$\mathcal{M}\downarrow$	$S_\alpha\uparrow$	$F_\beta\uparrow$	$E_\xi\uparrow$	$\mathcal{M}\downarrow$
f_1	.644	.453	.632	.157	.685	.559	.650	.174	.679	.497	.663	.147	.706	.572	.674	.143	.665	.472	.656	.151
f_1&l_s	.668	.519	.699	.125	.727	.675	.743	.138	.685	.537	.720	.121	.743	.681	.775	.107	.687	.547	.727	.121
f&l_c	.813	.725	.806	.075	.846	.810	.836	.090	.733	.597	.712	.103	.860	.820	.858	.065	.804	.691	.807	.086
Full	.828	.747	.859	.060	.860	.852	.883	.071	.791	.701	.816	.070	.890	.878	.919	.043	.810	.719	.838	.070

image prior information. We show the performance with a label "f_1 & l_s" in Table 6.2. We observe a performance improvement compared with "f_1", which indicates the usefulness of the edge-aware smoothness loss.

(3) Noisy-aware encoder-decoder without edge-aware smoothness loss: To study the effect of the smoothness regularization, we try to remove the smoothness loss from our model. As a result, we find that it will lead to trivial solutions i.e., , $S_i = \mathbf{0}_{H\times W}$ for all training images.

(4) Alternative smoothness loss: We also replace our smoothness loss l_s by a cross-entropy loss $l_c(S, X)$ that is also defined on the first-order derivative of the saliency map S and that of the image X. The performance is shown in Table 6.2 as "f & l_c", which is better than or comparable with the existing weakly supervised/unsupervised methods shown in Table 6.1. By comparing the performance of "f & l_c" with that of the full model, we observe that the smoothness loss $l_s(S, X)$ in Eq. 6.8

Table 6.3: Experimental results for model analysis. ↑ & ↓ denote larger and smaller is better, respectively.

	DUTS				ECSSD				DUT				HKU-IS				THUR			
Model	S_α ↑	F_β ↑	E_ξ ↑	$\mathcal{M}$ ↓	S_α ↑	F_β ↑	E_ξ ↑	$\mathcal{M}$ ↓	S_α ↑	F_β ↑	E_ξ ↑	$\mathcal{M}$ ↓	S_α ↑	F_β ↑	E_ξ ↑	$\mathcal{M}$ ↓	S_α ↑	F_β ↑	E_ξ ↑	$\mathcal{M}$ ↓
f-BAS	.870	.823	.894	.042	.910	.910	.935	.040	.839	.769	.866	.056	.904	.900	.945	.032	.821	.737	.840	.073
f-RBD	.824	.753	.854	.066	.869	.856	.890	.070	.776	.675	.799	.082	.886	.863	.918	.047	.803	.700	.823	.082
f-MR	.814	.759	.839	.064	.857	.856	.876	.073	.762	.669	.779	.079	.972	.866	.901	.050	.794	.696	.804	.086
f-GS	.787	.740	.811	.071	.826	.836	.843	.087	.737	.652	.753	.083	.837	.843	.865	.062	.804	.723	.840	.071
RBD	.644	.453	.632	.157	.685	.559	.650	.174	.679	.497	.663	.147	.706	.572	.674	.143	.665	.472	.656	.151
MR	.620	.442	.596	.199	.686	.567	.632	.191	.642	.476	.625	.191	.668	.545	.628	.180	.639	.460	.624	.179
GS	.619	.414	.623	.184	.657	.507	.622	.208	.637	.437	.633	.175	.690	.534	.660	.169	.636	.427	.634	.176
f_1*	.840	.769	.868	.054	.893	.883	.915	.054	.783	.676	.802	.073	.894	.871	.926	.040	.815	.720	.834	.077
f*	.861	.803	.887	.045	.906	.899	.927	.046	.815	.721	.836	.060	.905	.887	.933	.036	.831	.743	.849	.070
cVAE	.771	.695	.842	.078	.817	.812	.874	.086	.747	.665	.801	.085	.824	.800	.895	.068	.754	.659	.800	.100
Ours	.828	.747	.859	.060	.860	.852	.883	.071	.791	.701	.816	.070	.890	.878	.919	.043	.810	.719	.838	.070

works better than the cross-entropy loss $l_c(S, X)$. The former puts a soft constraint on their boundaries, while the latter has a strong effect on forcing both boundaries of S and X to be the same. Although the saliency boundary are usually aligned with the image boundary, but they are not exactly the same. A soft and indirect penalty for edge dissimilarity seems to be more useful.

6.4.4 Model Analysis

We further explore our proposed model in this section.

(1) Learn the model from saliency labels generated by fully supervised pre-trained models: One way to use our method is treating it as a boosting strategy for the current fully-supervised models. To verify this, we first generate saliency maps by using a pre-trained fully-supervised saliency network, e.g., BASNet [33]. We treat the outputs as noisy labels, on which we train our model. The performances are shown in Table 6.3 as f-BAS. By comparing the performances of f-BAS with those of BASNet in Table 6.1, we find that f-BAS is comparable with or better than BASNet, which means that our method can further refine the outputs of the state-of-the-art pre-trained fully-supervised models if their performances are still far from perfect.

(2) Create one single noisy label for each image: In previous experiments, our noisy labels are generated by handcrafted feature-based saliency methods in the setting of multiple noisy labels per image. Specifically, we produce three noisy labels for each training image by methods RBD [24], MR [26] and GS [183], respectively. As our method has no constraints on the number of generated noisy labels per image, we conduct experiments to test our models learned in the setting of one noisy label per image. In Table 6.3, we report the performances of the models learned from noisy labels generated by RBD [24], MR [26] and GS [183], respectively. We use f-RBD, f-MR and f-GS to represent their results, respectively. We observe comparable

performances with those using the setting of multiple noisy labels per image, which means our method is robust to the number of noisy labels generated from each image and the quality of the generated noisy labels. (RBD ranks the 1^{st} among unsupervised saliency detection models in [42]. RBD, MR and GS represent different levels of qualities of noisy labels). We also show in Table 6.3 the performances of the above handcrafted feature-based methods, which are denoted by RBD, MR and GS, respectively. The big gap between RBD/MR/GS and f-RBD/f-MR/f-GS demonstrates the effectiveness of our model.

(3) Train the model from clean labels: The proposed noise-aware encoder-decoder can learn from clean labels, because clean label can be treated as a special case of noisy label, and the noise generator will learn to output zero noise maps in this scenario. We show experiments on training our model from clean labels obtained from the DUTS training dataset. The performances denoted by f* are shown in Table 6.3. For comparison purpose, we also train the encoder-decoder component without the noise generator module from clean labels, whose results are displayed in Table 6.3 with a name f_1*. We find that (1) our model can still work very well when clean labels are available, and (2) f* achieves better performance than f_1*, indicating that even though those clean labels are obtained from training dataset, they are still "noisy" because of imperfect human annotation. Our noise-handling strategy is still beneficial in this situation.

(4) Train the model by variational inference: We train our model by alternating back-propagation algorithm that maximizes the observed-data log-likelihood, where we adopt Langevin Dynamics to draw samples from the posterior distribution $p_\theta(Z|Y,X)$, and use the empirical average to compute the gradient of the log-likelihood in Eq.(6.4). One can also train the model in a conditional variational inference framework [51] as shown in Eq. (6.9). Following cVAE [51], we design an inference network $p_\phi(Z|Y,X)$, which consists of four cascade convolutional layers and a fully connected layer at the end, to map the image X and the noisy label Y to the $d = 8$ dimensional latent space Z. The resulting loss function includes a reconstruction loss $\|Y_i - f(X_i, Z_i, \theta)\|^2$, a KL-divergence loss $\mathrm{KL}(p_\phi(Z|Y,X)\|p_\theta(Z|Y,X))$ and the edge-aware smoothness loss presented in Eq.(6.8). We present the cVAE results in Table 6.3. Our results learned by ABP outperforms those by cVAE. The main reason lies in the fact that the gap between the approximate inference model and the true inference model, i.e., , $\mathrm{KL}(p_\phi(Z|Y,X)\|p_\theta(Z|Y,X))$, is hard to be zero in practise, especially when the capacity of $p_\phi(Z|Y,X)$ is less than that of $p_\theta(Z|Y,X)$ due to an inappropriate architectural design of $p_\phi(Z|Y,X)$. On the contrary, our Langevin Dynamics-based inference step, which is derived from the model, is more natural and accurate.

6.5 Conclusion

Although clean pixel-wise annotations can lead to better performance, the expensive and time-consuming labeling process limits the applications of those fully supervised

models. Inspired by previous work [124, 66, 99], we propose a noise-aware encoder-decoder network for disentangled learning of a clean saliency predictor from noisy labels. The model represents each noisy saliency label as an addition of perturbation or noise from an unknown distribution to the clean saliency map predicted from the corresponding image. The clean saliency predictor is an encoder-decoder framework, while the noise generator is a non-linear transformation of a Gaussian noise vector, in which the transformation is parameterized by a neural network. Edge-aware smoothness loss is also utilized to prevent the model from converging to a trivial solution. We propose to train the model by a simple yet efficient alternating back-propagation algorithm [139, 234], which is superior to variational inference. Extensive experiments conducted on different benchmark datasets demonstrate the effectiveness and robustness of our model and learning algorithm. Note that, we choose VGG16-Net [67] backbone as it has less parameters than the ResNet backbones [28]. We observe that the capacity of the backbone network influences noisy labeling based models a lot. On the one hand, higher capacity means better regression/classification ability with the given training dataset. However, the higher capacity can also lead to overfitting issues, where the model may overfit on the noisy data, leading to poor generalization ability. We will investigate into this issue.

Chapter 7

Learning Saliency from Single Noisy Labelling: A Robust Model Fitting Perspective

In Chapter 6, we introduced a latent variable model to learn saliency from noisy labelling. With the same goal, we address a natural question: *can we learn saliency prediction while identifying clean labels in a unified framework given the noisy labels?* To answer this question, we call on the theory of **robust model fitting** and formulate deep saliency prediction from a single noisy labelling as robust network learning and exploit model consistency across iterations to identify inliers and outliers (i.e., , noisy labels). Extensive experiments on different benchmark datasets demonstrate the superiority of our proposed framework, which can learn comparable saliency prediction with state-of-the-art fully supervised saliency methods. Furthermore, we show that simply by treating ground truth annotations as noisy labelling, our framework achieves tangible improvements over state-of-the-art methods.

7.1 Introduction

Visual saliency prediction [27, 245] has enjoyed a great performance leap in recent years, thanks to the advances in deep Convolutional Neural Networks (CNNs) [138] and in particular through the use of fully supervised learning ([46, 33, 130, 34, 246, 215]). However, the superior performance of fully supervised saliency prediction models come at the expense of gathering large-scale annotated data [41, 42], which is extremely expensive and time-consuming. To minimize the labor associated with labelling, a viable direction is to learn deep models from noisy datasets [247].

In this chapter, we are interested in learning saliency prediction from *a single noisy labelling*, where the noisy labelling is produced by existing easy-to-access and cheap unsupervised conventional saliency methods [24, 189]. Under such a configuration, for each image, only one single noisy version of a saliency map is available. It is well-known that deep CNNs can memorize data even when the labels are completely random [244]. Therefore, directly training with noisy data may guide the network

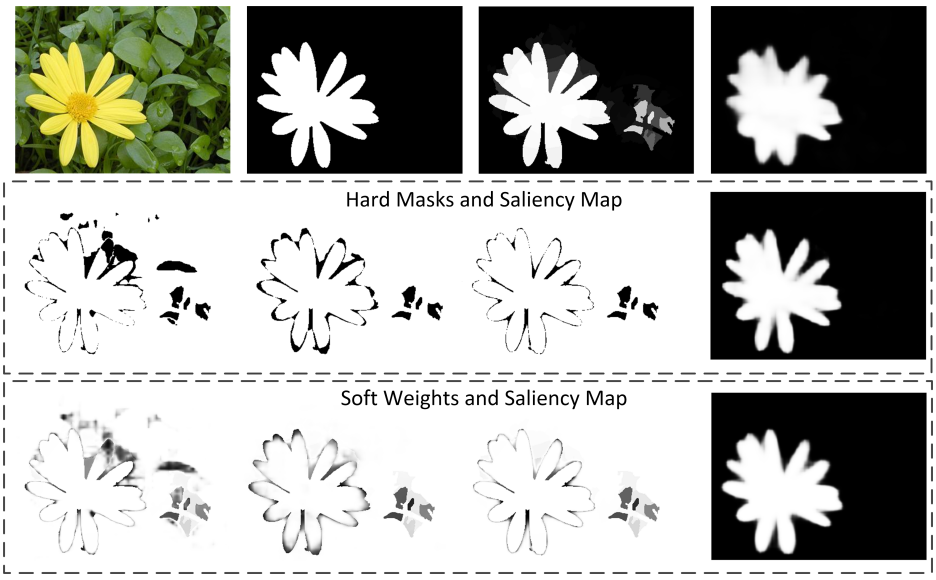


Figure 7.1: First row: input image, its ground truth saliency map, noisy saliency map by RBD [24] and saliency map by MNL [66]. Second and third rows show the early, middle and final iterations masks (or weights) and saliency maps of our hard mask selection and soft mask reweigting methods respectively.

to overfit to the corrupted labels. While considerable efforts have been made in learning deep models for image classification from noisy annotations [247, 248, 249, 118, 127, 247], the pixelwise dense labelling counterparts (e.g., saliency prediction [27], semantic segmentation [138], instance segmentation [250], and monocular depth estimation [251]) have received little attention.

Existing work in learning from dense noisy labelling [66, 124, 99] requires multiple noisy versions of pixel-wise labelling for each input image. Zhang *et al.* [66] made a strong assumption in modeling the noise with a single Gaussian distribution, which may hinder performance under complex noise distributions. The recursive optimization in [124] depends on a dedicated design and is computationally expensive. Nguyen *et al.* [99] defined image-level loss function to train with noisy labels to generate coarse saliency map, and then iteratively refined it with moving average and fully-connected CRF. Different from them, we intend to design a model without using a noise prior, and deal with learning from a single noisy labelling in a much more efficient way.

Specifically, we call on the theory of "robust model fitting" [252] based on one key observation: *though dense labelling is never free from noise, there exists a sufficient amount of clean and noise-free labels. Such clean labels, if correctly identified, can be used to achieve high-quality outcomes.* Thus we pose learning saliency from a single noisy labelling as "learn a deep saliency model and identify inliers and outliers in a unified framework".

The success of robust model fitting depends on the validation of two assumptions:

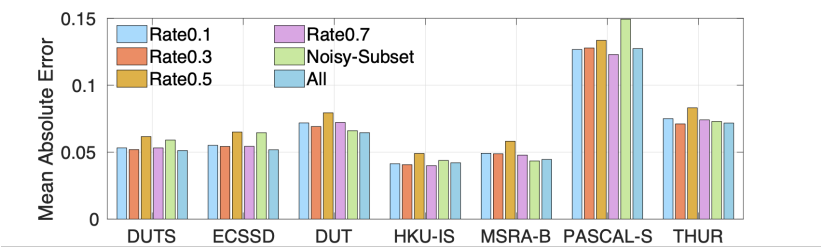


Figure 7.2: Performance evaluation (Mean Absolute Error) of saliency prediction by training a same network with different levels of supervision across seven different datasets.

1. A subset of samples is sufficient to fit the model;

2. Inliers are consistent with each other in reaching a consensus (given the desired model) while outliers are not.

To illustrate whether the first assumption holds, we trained a deep network with varying levels of supervision on a clean label dataset (see Fig. 7.2). We achieve this by randomly sampling a specific percentage of pixels of the prediction to compute the partial cross-entropy loss. In Fig. 7.2, "Rate0.1" means the percentage of sampling is 10%, "Noisy-Subset" indicates supervision as the intersection of noisy label and clean label[1], "All" is trained with pixel-wise clean dataset. The reasonable level of performance stability in Fig. 7.2 reinforces our belief that partial labels can lead to comparable saliency models. Furthermore, we find that the higher rates of sampling may not always lead to better performance. The main reason is that pixels contribute differently to the network training. For example, many local image regions in training sets may be of similar appearance which will be less helpful in network learning [253]. Also, we notice that in some datasets, the random sampling based model outperforms "All". This is because our training data is hand-annotated, and it may be poorly aligned, especially along object edges [254]. Further, we observe worse performance of the "Noisy-Subset" in the PASCAL-S dataset, which is the result of the relatively poor generalization ability with partial supervision. To prevent this, we introduce edge preserving saliency in Section 7.4.

To identify inliers, our idea is to benefit from consistency, a measure we define based on evolution of the network. In Fig. 7.1, we demonstrate how the network evolves across iterations, which reveals that: **1**) clean pixels inside or outside salient objects tend to have higher consistencies in model prediction; **2**) pixels with noisy labels or those on object boundaries vary across iterations have lower consistencies. This observation supports the second assumption, which is also consistent with observations in [123].

Our proposed method focuses more on pixels with high consistency. Such a mechanism allows us to find inliers (high confidence samples), similarly to RANSAC [252].

[1]We compare noisy label with clean label, and define the regions in the noisy label with accurate prediction as the "Noisy-Subset"

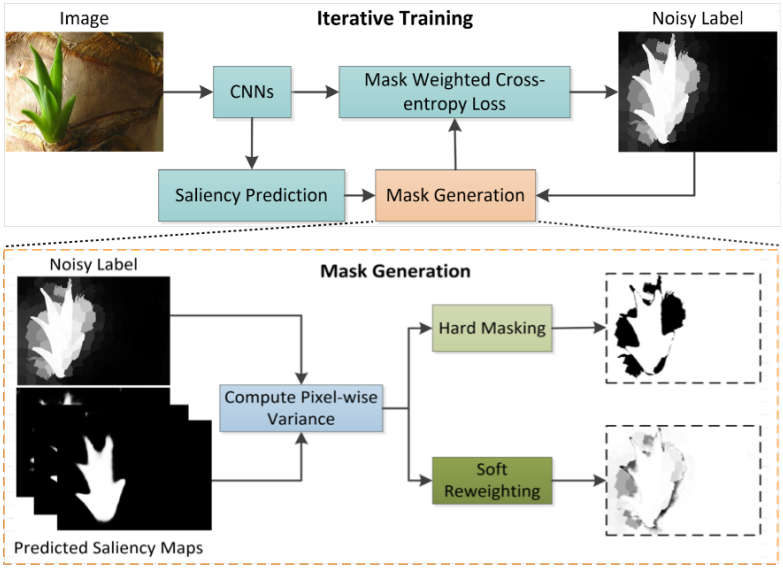


Figure 7.3: Conceptual illustration of our framework. We start with noisy pseudo label. By iteratively updating a dense mask for partial supervision with the noisy labelling, we can effectively identify the inlier labels inside the noisy labels.

We propose two simpler and more effective approaches by iteratively updating a dense mask to find or weight the possible inliers, where each mask element represents the weight of the current pixel to contribute to the loss. As shown in Fig. 7.1, after several iterations, our method successfully identifies inliers and ourliers[2]

Our main contributions can be summarized as:

1. We pose deep saliency prediction from a single noisy labelling as a robust model fitting problem, and jointly learn a deep saliency model and identify inliers as shown in Fig. 7.3.

2. We introduce two approaches, namely hard mask selection and soft mask reweighting, to assign each pixel a hard $\{0,1\}$ mask or soft $[0,1]$ weight for loss updating, representing our confidence on noisy level of this pixel.

3. Experimental results show that our method outperforms existing weakly supervised or unsupervised methods, and achieves comparable performances with state-of-the-art deep fully supervised saliency prediction methods.

7.2 Related Work

In this section, we briefly review three forms of deep saliency prediction methods, namely supervised, semi-supervised and unsupervised methods. Interested readers

[2]In our saliency prediction task, inliers represent noise-free label/annotation while outliers indicate incorrect label/annotation.

are referred to surveys such as [27, 245, 153] for more detailed discussions. We later discuss related studies in learning from noisy labels.

Supervised Saliency Prediction: Fully supervised deep saliency prediction models [181, 31, 32, 34, 66, 46, 33, 130, 215, 255, 256] train classifiers to assign saliency value to each pixel (or superpixel). Wei *et al.* [34] presented cross feature module and cascaded feedback decoder to effectively fuse different level feature, and a position aware loss was further introduced to emphasize hard pixels during training. Wu *et al.* [130] introduced stacked cross refinement network to generate edge-preserving features for accurate saliency detection. To better use context information, Liu *et al.* [31] presented a pixel-wise context attention strategy to use effective neighbour pixels for saliency prediction. Liu *et al.* [46] expanded the role of pooling in convolutional neural networks by building a global guidance module and a feature aggregation module for detail enriched saliency prediction. Qin *et al.* [33] introduced a hybrid loss for boundary-aware saliency detection to generate accurate saliency prediction with sharp boundaries.

Weakly Supervised Saliency Prediction: Weakly supervised saliency models [41, 98, 63, 236] learn saliency from cheaper annotations. Among these models, Wang *et al.* [41] introduced a foreground inference network to iteratively produce potential saliency maps using image-level labels. Li *et al.* [236] designed contour-to-saliency network by converting an existing contour detection model to saliency detection model without using manual saliency annotations. Zhang *et al.* [63] labeled scribble saliency dataset to learn saliency from scribble annotation.

Unsupervised Saliency Prediction: Different from weakly supervised saliency models, where human annotation is still needed. The unsupervised saliency models learn saliency without human annotation. Zhang *et al.* [124] proposed to fuse noisy saliency maps from handcrafted feature based methods with heuristic. Zhang *et al.* [66] formulated unsupervised saliency detection as learning from multiple noisy labelling, where a latent saliency prediction module and a noise modeling module work collaboratively and are optimized jointly. Nguyen *et al.* [99] defined image-level loss function to train with multiple noisy label to generate coarse saliency map, and then iteratively refined it with moving average and fully-connected CRF. In contrast to previous studies [124, 66, 99], our proposed approach requires only one single noisy label per pixel to learn saliency. Also, our method does not rely on any predefined noise distribution priors [66].

Learning from Noisy Labels To handle noisy labels, three main directions have been explored: 1) developing regularization techniques [113]; 2) estimating the noise distribution [114]; 3) training on selected samples [115]. Since our model belongs to the third category, we focus on sampling-based methods. Jiang *et al.* [118] proposed to learn a MentorNet to produce a curriculum for the StudentNet. Thus the latter one can focus on samples where their labels are possibly correct. Liu and Tao [120] presented an importance reweighting method, where uncertainty introduced by classification noise is reduced by estimating an importance weight parameter. Ren *et al.* [122] learned to assign weights to training examples based on their gradient directions during each mini-batch. Nguyen *et al.* [123] introduced self-ensemble label

filtering to progressively filter out wrong labels during training.

All of the above learning from noisy labels methods deal with image classification. In this chapter, we propose a principled method for dealing with the dense prediction task of saliency prediction from the robust model fitting perspective. Different from [127], where labels are updated according to the outputs of each iteration, we fix the noisy labels and make use of the inlier/outlier mask to identify the clean ones. To the best of our knowledge, this is the first time that sampling from single noisy labelling has been used to address the task of saliency prediction.

7.3 Model

We focus on learning saliency from a single noisy labelling by using a deep neural network. Specifically, given a color image $\boldsymbol{x}_i$, we would like to learn a saliency map $\boldsymbol{y}_i$ from its noisy saliency map $\hat{\boldsymbol{y}}_i$, which is produced by a "cheap" and easy to access unsupervised saliency method. Directly training the network with a single noisy labelling will not work as it is well-known that network training is highly prone to noise in the supervision signals [244]. Existing multiple noisy labelling based methods [124, 66, 99] also will not work due to the requirement of multiple noisy labelling for label updating [124, 99] or noise distribution estimation [66]. Instead, we propose a principled way to infer saliency maps from a robust model-fitting perspective, thus we can simultaneously infer the saliency map robustly and identify the inliers for the desired model.

As mentioned before, the success of robust model fitting depends on fulfilling two assumptions as described in Section 7.1. In the following sections, we formulate the problem and further study the validity of the above assumptions.

7.3.1 Problem Formulation

We start with a training set $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N$, where each $\boldsymbol{x}_i$ is a color image of size $U \times V$ and $\boldsymbol{y}_i \in \{0,1\}^{U \times V}$ is a binary saliency map (also denoted by $\{(\boldsymbol{x}_i, \hat{\boldsymbol{y}}_i)\}$ in the noisy labelling setting). Deep saliency models learn a mapping function $f_\Theta : \mathbb{R}^{U \times V \times 3} \to [0,1]^{U \times V}$, where Θ is a set of network parameters. Therefore, $f_\Theta(\boldsymbol{x}_i) = \boldsymbol{s}_i$ denotes the predicted saliency map, and the empirical risk when learning from clean labelling can be defined as follows:

$$\mathcal{L}(\Theta \mid \boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{N} \sum_{i=1}^{N} \sum_{(u,v)} \ell \left(\boldsymbol{s}_i^{(u,v)}, \boldsymbol{y}_i^{(u,v)} \right), \tag{7.1}$$

where $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^N, \boldsymbol{Y} = \{\boldsymbol{y}_i\}_{i=1}^N$, and (u,v) denotes the pixel coordinates in an image, and $\ell : [0,1] \times \{0,1\} \to \mathbb{R}$ is the cross-entropy loss defined as:

$$\ell(s, y) = -\big(y \log(s) + (1-y) \log(1-s)\big) . \tag{7.2}$$

When clean labels are available, the optimal network model is obtained by minimizing Eq. 7.1 using stochastic gradient descent.

7.3.2 Robust Model Fitting Perspective

For a dense prediction task, the network is trained by minimizing the per-pixel wise defined loss function as shown in Eq. 7.1. We argue that such dense information is mostly redundant for our problem and propose to exploit the robust model fitting principle in the context of deep learning. Even though the noisy labels come from weak saliency model (an unsupervised one), we assume there still exists a considerable proportion of labels that are correct, and can be considered as inliers to the desired model. With this, we are left with two problems: 1) Whether all the labelling is needed to supervise network learning? 2) How to identify the inliers under deep convolutional neural network?

Whether all the labelling is needed? The loss function for learning dense saliency prediction (i.e., , Eq. 7.1) is basically a sum of per-pixel loss and there is no pairwise or high-order terms in the loss function. Therefore, evaluating the loss function with partial supervision will not change the learning procedure. To prove this assumption/observation, we train the same deep saliency prediction network with different levels of supervision by random sampling $r\%$ of image $\boldsymbol{x}_i$ and clean label $\boldsymbol{y}_i$ for loss function evaluation. As illustrated in Fig. 7.2, when we varied the proportion of available labels from 10% to 100%, the performance of saliency prediction is relatively stable with different levels of supervision, which indicates that partial cross-entropy [111] is indeed sufficient to train a reasonable saliency detection model.

How to identify inliers/outliers? Now, we are left with the problem of how to identify the inliers to the desired saliency prediction model. Existing methods use the fitting loss as a function to determine inliers [118][252]. This principle has also been used in meta-learning and self-paced learning [249]. To be more specific, an inlier is determined as samples with smaller loss: $\mathbb{1}\left(\ell(f_\Theta(\boldsymbol{x}_i), \hat{\boldsymbol{y}}_i) \leq \lambda\right)$, where the hyperparameter λ is a user-defined threshold. The above strategy could be sensitive to the initialization due to its iterative optimization style. We instead propose to exploit the *model consistency during iterations*, and use uncertainty/variance as guidance to iteratively select pixels for next iteration. Specifically, we propose to score the pixels with the network evolution consistency, where the underlying assumption/belief is that the network learns to predict inliers consistently in the initial phases of network evolution, and eventually learns to fit the outliers [123].

In the active learning setting, the prediction variance can be used to measure the uncertainty of each sample (pixel) for either a regression problem or a classification problem [257], where high variance stands for uncertain samples, and low variance represents consistent model behavior. We select low uncertainty/variance samples for model updating. Thus we introduce a dense weight matrix Ω_i for image $\boldsymbol{x}_i$, with each element in Ω_i representing the weight of the current pixel for loss updating. Ω_i is then defined as:

$$\Omega_i^{(u,v)} = g(H_i(u,v)), \tag{7.3}$$

where $H_i = \cup_t H_i^{(t-1)}$ is the set (or union $\cup_t$) of predictions for image $\boldsymbol{x}_i$ before the current iteration t, and function g defines the sampling strategy given the history of predictions, which represents an indicator function for hard mask selection and a

linear function for soft mask reweighting[3].

Specifically, we formulate the problem as joint optimization of network parameters Θ to predict the saliency map, and masking parameters Ω to identify inliers to the network model as follows:

$$(\Theta^*, \Omega^*) = \arg\min_{\Theta,\Omega} \mathcal{L}(\Theta, \Omega \mid \boldsymbol{X}, \hat{\boldsymbol{Y}}) + \mathcal{R}(\Omega, \lambda), \tag{7.4}$$

where $\hat{\boldsymbol{Y}}$ is the noisy label set, function $\mathcal{R}(\cdot,\cdot)$ denotes a sampling strategy with regularization to maximize the inlier set of the desired saliency prediction model (either in a hard way or in a soft way) and λ is a sampling related hyperparameter. The loss function $\mathcal{L}(\Theta, \Omega \mid \boldsymbol{X}, \hat{\boldsymbol{Y}})$ (denoted as $\mathcal{L}(\Theta, \Omega)$ for convenience) with the sampling strategy controlled by masking Ω is defined as:

$$\mathcal{L}(\Theta, \Omega) = \frac{1}{N} \sum_{i=1}^{N} \sum_{(u,v)} \Omega_i^{(u,v)} \; \ell\left(\boldsymbol{s}_i^{(u,v)}, \hat{\boldsymbol{y}}_i^{(u,v)}\right) \; . \tag{7.5}$$

We investigate two different ways of generating $\{\Omega_i\}_{i=1}^N$. With the hard mask pixel sampling strategy, we have $\Omega_i^{(u,v)} \in \{0,1\}$, thereby selecting a subset of pixels to evaluate the loss function. With the soft mask pixel reweighting strategy, we have $\Omega_i^{(u,v)} \in [0,1]$, assigning soft weights to each pixel, which can be viewed as a relaxed version of the first one.

7.4 Solutions

In this section, we present the above mentioned two solutions to our proposed model for learning saliency from single dense noisy labelling. We introduce a hard masking strategy in Sec. 7.4.1 and a soft reweighting strategy in Sec. 7.4.2.

7.4.1 Hard Mask Selection

In the context of our problem, stochastic optimizers such as stochastic gradient decent (SGD) can be seen as a sampler where each pixel is assigned the same constant weight for loss updating. To reduce the effect of noisy labelling, one possible way is to decrease the weight of difficult examples during the course of training, making the model more robust to the noise [258, 259, 115]. Along this pipeline, we propose a hard mask selection method to learn from single noisy labelling. To this end, we make use of the pixel selection strategy $\Omega_i^{(u,v)} \in \{0,1\}$ and set $\Omega_i^{(u,v)} = 1$ if (u,v) is a clean pixel of the i-th image and 0 otherwise.

With the loss defined in Eq. 7.5, and an initial sampling strategy Ω, alternating minimization [260] can be used to update the network parameters Θ and inlier selection strategy Ω for the next iteration. Thus the loss function is alternatively minimized with respect to Θ and Ω, one at a time while the other is fixed. Given

[3]Note that, our framework facilitates computation of supervised loss mostly on possible clean pixels rather than the entire set including noisy labels.

a sampling strategy Ω, the network parameter Θ is typically optimized by SGD. For a fixed network parameter Θ, following [260], we exploit the "inlier set maximization" principle in robust model fitting and define the regularization $\mathcal{R}(\Omega, \lambda)$ as: $\mathcal{R}(\Omega, \lambda) = -\lambda \sum_{i=1}^{N} \|\Omega_i\|_1$, i.e., , the (relaxed) number of inliers, then the optimal Ω_i can be derived as:

$$\Omega_i^{(u,v)} = \mathbb{1}(\ell(\boldsymbol{s}_i^{(u,v)}, \hat{\boldsymbol{y}}_i^{(u,v)}) \leq \lambda), \tag{7.6}$$

where $\mathbb{1}$ is the indicator function. This inlier selection strategy could be sensitive to initialization due to its iterative optimization style. Instead, we propose to exploit the network model consistency during iterations, and use uncertainty/variance as guidance to select samples that are probably clean. Thus we score the samples with the network evolution consistency. Also, according to [115, 261], prediction variance is proportional to network loss, and larger prediction variance indicates a high uncertainty sample. Thus, we replace Eq. 7.6 with:

$$\Omega_i^{(u,v)} = \mathbb{1}(\text{var}_{u,v}(H_i) \leq \lambda), \tag{7.7}$$

where $\text{var}_{u,v}(H_i)$ represents the variance at image point (u, v) of the prediction for all history models.

Eq. 7.7 suggests selecting pixels that produce smaller variance, or as [260] defined, the "easy" samples. For a fixed Θ, those easy samples with variance smaller than threshold λ will be selected for training in the next iteration. Otherwise, they will not be selected. For a fixed Ω, the model is only trained on selected pixels to update Θ (partial cross-entropy loss [111]). The hyper-parameter λ controls the learning pace, especially the amount of samples for later iteration training, we use $\lambda = 0.02$ in all the hard masking related experiments.

Initialization: Within the hard mask selection strategy, we randomly select a small proportion (10% to achieve stable training) of the noisy labelling to get an initial Ω. An example of the evolution of mask is illustrated as a binary image in the first, second and third image of the second row in Fig. 7.1. The experimental results show that we can achieve relatively stable performance with different initialization of the hard mask.

7.4.2 Soft Mask Reweighting

The hard mask selection strategy can be interpreted as substituting the original cross-entropy loss with a truncated loss [116], where samples with weak consistency over the history of models are assigned constant loss, which leads to zero loss gradient. Thus these samples will not contribute to the learning dynamics.

However, the hard mask pixel selection method has the potential to discard possibly clean but diversely behaved samples during training. Thus, we further propose a soft mask reweighting strategy to assign a soft weight for each pixel in an image. The loss function $\mathcal{L}_{\text{soft}}(\Theta, \Omega)$ is then defined as in Eq. 7.5 where the dense weight

matrix Ω_i is defined as

$$\Omega_i^{(u,v)} = 2/(1 + \exp(k \ \mathrm{var}_{u,v}^2(H_i))), \tag{7.8}$$

where $k \in \mathbb{R}^+$ is used to control the descent degree of the soft weights according to different variance. We set $k = 4$ in our experiments. The soft weight Ω_i encourages pixels with consistent behavior to contribute more than those with diverse behavior.

Note that, even though these two strategies are discussed separately, they can generalize to a uniform model with k set to a large value to get an approximate hard mask selection model.

Initialization: Within the soft mask reweighting strategy, in the first iteration, we set all the pixels as clean, leading to an all-1 matrix Ω. Iteratively, we get a gray weight matrix (in the range of $[0, 1]$) as shown in the first, second and third image of the third row in Fig. 7.1.

7.4.3 Edge Preserved Saliency Detection

Both hard mask selection and soft mask reweighting strategies iteratively assign pixel-wise importance to each sample to update the loss, which may lead to a saliency map with a blurred boundary, especially for the sampling based hard mask selection strategy. To recover the lost structure information, we add another structure-aware regularization $\mathcal{R}_e$ similar to [262] to loss function in Eq. 7.5, which is defined as:

$$\mathcal{R}_e = \ell(e(\boldsymbol{x}), e(\boldsymbol{s})), \tag{7.9}$$

where $e(.)$ is an edge detection operator[4], $\boldsymbol{x}$ and $\boldsymbol{s}$ are the input color image and predicted saliency map, and ℓ is the cross-entropy loss defined as Eq. 7.2. The basic assumption of Eq. 7.9 is that the edge of the saliency map should be similar to the edge of the raw input image. Thus, the edge-preserved loss function is defined as: $\mathcal{L}_e(\Theta, \Omega) = \mathcal{L}(\Theta, \Omega) + \mathcal{R}_e$, where $\mathcal{L}(\Theta, \Omega)$ is defined in Eq. 7.5.

7.4.4 Model Analysis

1) How to handle error propagation? In each iteration of our method, we select a subset of possible clean pixels through pseudo labelling to train the model, and it is possible that we may wrongly select some noisy pixels. To prevent possible error propagation, we adopt an early stopping strategy, since deep models tend to learn clean pixels before over-fitting to noisy samples.

2) How are the hard mask and soft mask connected? Considering both Eq. 7.7 and Eq. 7.8, we discover that the soft mask reweighting solution can be treated as a general case of the hard mask solution, and by setting k in Eq. 7.8 to a large number, the soft mask in Eq. 7.8 is approximately a hard mask as in Eq. 7.7. **3) How dose the model perform during testing?** Our model is trained with noisy label as supervision. During testing, given RGB image $\boldsymbol{x}$, we directly produce its saliency map $\boldsymbol{s}$.

[4] $e(.)$ is achieved by extracting first-order derivative of the image.

7.5 Experimental Results

7.5.1 Implementation

We trained our model using Pytorch, with 30 maximum epochs. We initialized our model using ResNet50 [28] trained for image classification, and adapt it to saliency prediction following [29][5]. We used the SGD method with momentum 0.9. The base learning rate is initialized as 2.5×10^{-4} with the "poly" decay policy. The whole training took 6 hours with training batch size 20 on a PC with an NVIDIA GeForce RTX GPU.

7.5.2 Setup

Dataset: We have evaluated the performance of our proposed model on eight saliency benchmarking datasets. We used 10,553 images from the DUTS dataset [41] for training and generated noisy saliency maps by using handcrafted feature based method (RBD [24] in particular, due to its efficiency). Testing images include: 1) the DUTS testing dataset (5,019); 2) ECSSD (1,000) [189], 3) DUT (5,168) [26], 4) HKU-IS (4,447) [30], 5) the MSRA-B Testing dataset (2,500) [9], 6) PASCAL-S (850) [229], 7) THUR (6,232) [190] and 8) SOC (1,200) [191], with number after dataset indicating the size of the corresponding dataset.
Evaluation metrics: Four evaluation metrics are used, including Mean Absolute Error ($\mathcal{M}$), mean F-measure (F_β), mean E-measure (E_ξ) [173] and S-measure (S_α) [172]. MAE is defined as the average per-pixel difference between the ground truth and the estimated saliency map. F-measure F_β is a region-based similarity metric. E-measure E_ξ combines local pixel values with the image-level mean value in one term, jointly capturing image-level statistics and local pixel matching information. S-measure S_α is a structure based measure, which combines both the region-aware and object-aware structural similarity metric.

7.5.3 Comparison with the State-of-the-Art

Quantitative Comparison: We compared our method with the state-of-the-art methods, and the results are reported in Table 7.1 and Fig. 7.4. "Our Models" in Table 7.1 indicate the performance of our methods, including 1) "HS" and 2) "SR" representing results of our proposed hard mask selection strategy and soft reweighting strategy respectively. "Ours" in Fig. 7.4 represents the results of our model using the soft reweighting strategy.

Our method starts with a single noisy labelling, and by assigning a weight/mask to each pixel, we adaptively select probably clean pixels of each training image for loss updating. Different from our approach, WSI [124] adopts an extra classification network with an image-level label [223], and a fully connected CRF [29] was also used for saliency map post-processing. Our methods achieve consistently better performance, with approximate 8% increase in both F_β and E_ξ with only single noisy

[5]Note that, we adopt the DeeplabV2[29] framework, and our solutions can be extended to other structures with ease.

Table 7.1: Benchmarking results. Bold numbers represent the best performance. ↑ & ↓ denote larger and smaller is better, respectively.

		Fully Sup. Models									Weakly Sup./Unsup. Models						Our Models	
	Metric	F3Net	NLDF	PoolNet	BASNet	AFNet	MSNet	SCRN	EGNet	MINet	DRFI	RBD	SBF	WSI	WSS	MNL	HS	SR
		[34]	[32]	[46]	[33]	[40]	[39]	[130]	[215]	[255]	[18]	[24]	[124]	[98]	[41]	[66]		
DUTS	$S_\alpha \uparrow$	**.888**	.816	.887	.866	.867	.862	.885	.887	.884	.676	.644	.743	.697	.748	**.813**	.793	.806
	$F_\beta \uparrow$	**.852**	.757	.840	.823	.812	.792	.833	.839	.844	.481	.453	.622	.569	.633	.725	.712	**.732**
	$E_\xi \uparrow$	**.920**	.851	.910	.896	.893	.883	.900	.907	.917	.628	.632	.763	.690	.806	.853	.853	**.853**
	$\mathcal{M} \downarrow$	**.035**	.065	.037	.048	.046	.049	.040	.039	.037	.155	.157	.107	.116	.100	.075	.070	**.067**
ECSSD	$S_\alpha \uparrow$	.919	.870	.919	.910	.907	.905	**.920**	.918	.920	.730	.685	.830	.805	.808	.846	**.855**	.852
	$F_\beta \uparrow$	**.921**	.871	.913	.913	.901	.886	.910	.914	.920	.589	.559	.798	.762	.774	.810	.844	**.863**
	$E_\xi \uparrow$	**.943**	.896	.938	.938	.929	.922	.933	.937	.945	.649	.650	.848	.792	.801	.836	**.904**	.885
	$\mathcal{M} \downarrow$	**.036**	.066	.038	.040	.045	.048	.041	.041	.036	.172	.174	.089	.114	.106	.090	**.066**	.069
DUT	$S_\alpha \uparrow$	**.839**	.770	.831	.836	.826	.809	.837	.841	.833	.696	.679	.747	.759	.730	.733	**.771**	.766
	$F_\beta \uparrow$	.766	.683	.748	**.767**	.743	.710	.749	.760	.757	.507	.497	.612	.641	.590	.597	.668	**.679**
	$E_\xi \uparrow$	.864	.798	.848	**.865**	.846	.831	.847	.857	.860	.653	.663	.763	.761	.729	.712	**.818**	.805
	$\mathcal{M} \downarrow$	**.053**	.080	.054	.057	.057	.064	.056	.053	.056	.150	.147	.108	.100	.110	.103	.080	**.072**
HKU-IS	$S_\alpha \uparrow$	.917	.879	**.919**	.909	.905	.907	.916	.918	.919	.740	.706	.829	.808	.822	.860	.854	**.863**
	$F_\beta \uparrow$	**.910**	.871	.903	.903	.888	.878	.894	.902	.909	.594	.572	.783	.763	.773	.820	.828	**.864**
	$E_\xi \uparrow$	**.952**	.914	.945	.943	.934	.930	.935	.944	.952	.669	.674	.855	.800	.819	.858	.906	**.909**
	$\mathcal{M} \downarrow$	**.028**	.048	.030	.032	.036	.039	.034	.031	.029	.145	.143	.075	.089	.079	.065	.054	**.047**
MSRA-B	$S_\alpha \uparrow$	-	.910	-	**.920**	.906	-	-	-	-	.813	.793	.880	.868	.853	**.889**	.880	.889
	$F_\beta \uparrow$	-	.869	-	**.905**	.883	-	-	-	-	.682	.688	.841	.836	.803	.867	.870	**.878**
	$E_\xi \uparrow$	-	.915	-	**.944**	.926	-	-	-	-	.734	.752	.895	.865	.849	.901	.913	**.915**
	$\mathcal{M} \downarrow$	-	.050	-	**.034**	.043	-	-	-	-	.123	.112	.060	.068	.078	.053	.050	**.048**
PASCAL-S	$S_\alpha \uparrow$	.802	.756	**.806**	.785	.797	.794	.801	.798	.795	.624	.614	.732	.669	.704	**.768**	.753	.765
	$F_\beta \uparrow$	**.833**	.793	.833	.821	.824	.813	.820	.827	.825	.510	.530	.735	.653	.698	.748	.773	**.774**
	$E_\xi \uparrow$	**.836**	.783	.833	.821	.827	.822	.821	.824	.830	.554	.582	.746	.647	.690	.741	**.768**	.762
	$\mathcal{M} \downarrow$	**.111**	.145	.114	.122	.116	.119	.118	.121	.115	.256	.247	.167	.206	.184	.158	**.148**	.153
THUR	$S_\alpha \uparrow$	**.838**	.801	-	.823	.825	.819	.845	-	-	.705	.665	.757	-	.775	.804	.794	**.806**
	$F_\beta \uparrow$	**.761**	.711	-	.737	.733	.718	.758	-	-	.512	.472	.627	-	.653	.691	.695	**.725**
	$E_\xi \uparrow$	**.858**	.827	-	.841	.840	.829	.858	-	-	.663	.656	.770	-	.775	.807	.816	**.839**
	$\mathcal{M} \downarrow$	**.066**	.081	-	.073	.072	.079	.066	-	-	.147	.151	.107	-	.097	.086	.086	**.072**
SOC	$S_\alpha \uparrow$	.828	.816	-	.841	.700	-	.838	**.850**	-	.709	.701	.763	-	.775	.814	.834	**.843**
	$F_\beta \uparrow$	.340	.319	-	.359	.062	-	**.363**	.346	-	.187	.212	.273	-	.271	.310	.321	**.344**
	$E_\xi \uparrow$	.846	.837	-	.864	.684	-	.859	**.866**	-	.689	.703	.774	-	.780	.823	.843	**.863**
	$\mathcal{M} \downarrow$	.098	.106	-	.092	.115	-	.099	**.085**	-	.197	.206	.153	-	.141	.093	.089	**.088**

labeling used. MNL [66] achieves comparable performance with some of the deep fully supervised models, while our method, using only a single noisy labelling (MNL relies on multiple noisy labels) is much cheaper, and its performance is also better, with on average of 3% performance increase of F_β and E_ξ, as well as around 2% decrease in $\mathcal{M}$. Figure 7.4 shows the mean F-measure (first row) and mean E-measure (second row) curves of our method and competing methods on four benchmark datasets. By iteratively identifying both inliers and ourliers with an edge-preserved regularization in Eq. 7.9, we achieve better performance compared with competing setting, and especially for the THUR dataset, our performance is even comparable with state-of-the-art fully supervised methods.

Qualitative Comparisons: Figure 7.5 demonstrates three visual comparisons. The

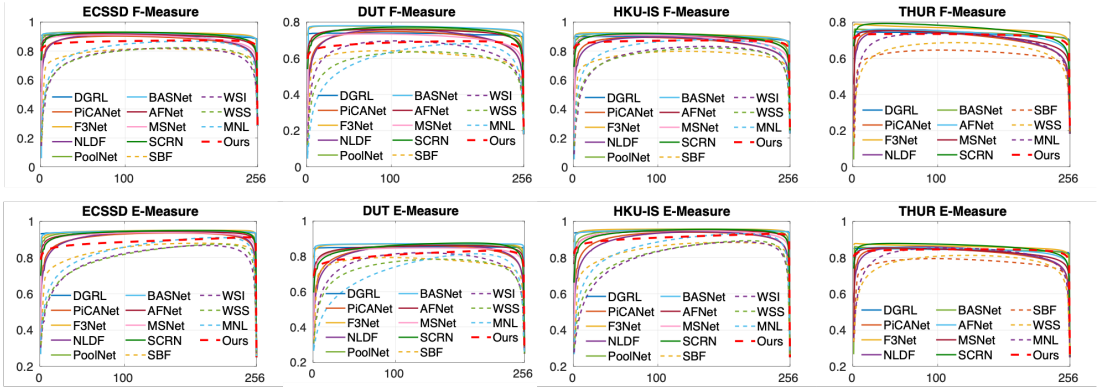


Figure 7.4: F-measure and E-measure curves on four benchmark datasets (ECSSD, DUT, HKU-IS and THUR). Best Viewed on screen.

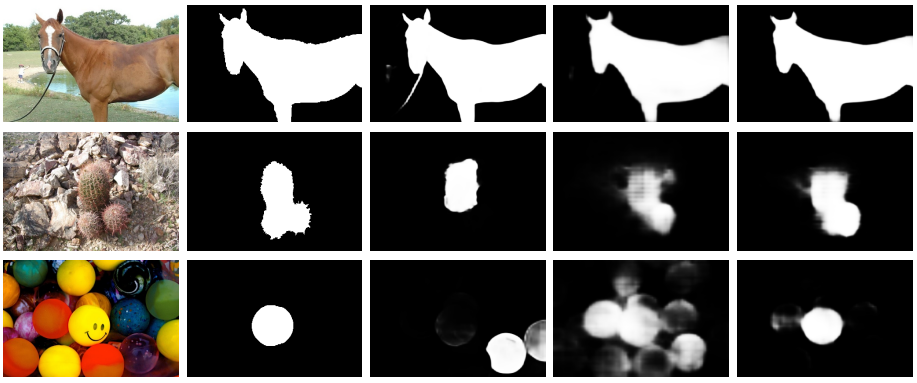


Figure 7.5: Comparison of saliency maps with competing methods.

salient object in the first row is large, which is a hard scenario for some deep models, while our model can produce high quality saliency maps with the complete salient region being segmented. The salient objects in the second image share similar appearance as the background, while our method can still produce saliency map with higher inter-class distinction. The background in the third image is quite complex, and competing methods (F3Net [34] and MNL [66]) fail to discriminate salient objects from the cluttered background, while our method produces nearly clear salient map.

7.5.4 Assumption Validation

It has been proved that a deep neural network can memorize any data even if it is highly corrupted [244], which makes it very sensitive to label noise, as shown in Table 7.2 "Noisy", representing training a neural network directly using noisy label as pseudo ground truth. Further, as illustrated in "NoS", with subset of clean labels selected from the noisy labels, performance can be boosted substantially. Beside this, for dense prediction tasks, all the pixels inside the image are used for loss computation. As discussed in Sec. 7.3.2, by randomly sampling a subset clean pixels for updating the loss, we achieve relatively stable performance as shown in Fig. 7.2. For

Table 7.2: Assumption Validation and Ablation Study

	Metric	DUTS	ECSSD	DUT	HKU-IS	MSRA-B	PASCAL-S	THUR
		Assumption Validation						
Noisy	$F_\beta \uparrow$	.5162	.6730	.5166	.6667	.7589	.6381	.5388
	$\mathcal{M} \downarrow$	.1493	.1454	.1490	.1220	.0941	.2173	.1426
NoS	$F_\beta \uparrow$	.7385	.8559	.6882	.8656	.8755	.7626	.7209
	$\mathcal{M} \downarrow$	.0656	.0730	.0702	.0475	.0487	.1562	.0717
		Ablation Study						
MR-O	$F_\beta \uparrow$	.4419	.5672	.4763	.5450	.6791	.5142	.4595
	$\mathcal{M} \downarrow$	.1989	.1913	.1907	.1798	.1323	.2658	.1792
MR-W	$F_\beta \uparrow$	.7162	.8614	.6639	.8498	.8756	.7749	.7094
	$\mathcal{M} \downarrow$	.0703	.0688	.0758	.0517	.0493	.1510	.0751
VGG	$F_\beta \uparrow$	.7392	.8635	.6825	.8626	.8706	.7907	.7124
	$\mathcal{M} \downarrow$	.0653	.0690	.0727	.0479	.0499	.1461	.0761
NoE	$F_\beta \uparrow$	.7177	.8499	.6603	.8501	.8686	.7666	.7174
	$\mathcal{M} \downarrow$	.0697	.0725	.0760	.0509	.0518	.1536	.0734
Clean	$F_\beta \uparrow$	.7689	.8826	.6764	.8707	.8631	.8075	.7197
	$\mathcal{M} \downarrow$	.0537	.0535	.0726	.0403	.0504	.1232	.0774
C_O	$F_\beta \uparrow$	.7799	.8863	.7027	.8703	.8788	.8035	.7325
	$\mathcal{M} \downarrow$	.0512	.0518	.0646	.0421	.0446	.1274	.0718
BAS_O	$F_\beta \uparrow$	.7922	.8918	.7456	.8958	.9031	.7995	.7470
	$\mathcal{M} \downarrow$	.0565	.0591	.0600	.0394	.0380	.1448	.0666
Web	$F_\beta \uparrow$	.7125	.7770	.6248	.7794	.8412	.6808	.6694
	$\mathcal{M} \downarrow$	.0829	.0936	.0866	.0863	.0619	.1796	.0899
Ours	$F_\beta \uparrow$	.7318	.8631	.6792	.8640	.8782	.7744	.7250
	$\mathcal{M} \downarrow$	.0674	.0687	.0723	.0473	.0482	.1529	.0719

some datasets, these sampling models achieve even better performance, which can be explained based on [263]. [263] proved that models can sometimes achieve lower generalization error after being trained with a subset of actively selected training data. In other words, focusing on informative samples can be beneficial even when all labels are available.

7.5.5 Ablation Study

We conduct different ablation studies to illustrate the effectiveness of our proposed method as shown in Table 7.2. We use the soft mask reweighting method for all the ablation experiments, and "Ours" in Table 7.2 is performance of the soft mask reweighting solution.

1) Different noisy labelling: In our experiments, we used RBD [24] to generate noisy labels. We further carried out an experiment with noisy saliency maps generated by MR [26], and show the performance as "MR-O" and "MR-W", representing the original performance of MR and learning saliency from MR with our solution. The big gap between "MR-O" and "MR-W" further indicates the effectiveness of our method. Note that, as our noisy label is generated by conventional method(s), we cannot control the actual portion of noisy data as the image-level noisy label handling

methods (they can control portion of noise as their noisy label is generated by adding noise to clean data), and we can only achieve this by using different noisy map generation methods.

2) Different network structure: We adopted ResNet50 as our backbone. To illustrate the superiority of our method regarding different network structure, we trained with VGG-16 [67] instead[6]. The performance reported as "VGG" clearly shows that the proposed method can still work well, which further illustrate the effectiveness of our noise handling strategy.

3) Learning without edge loss: As discussed in Sec. 7.4.3, training directly with the loss function in Eq. 7.5 will lead to a saliency map with blurred boundaries. We remove the edge loss $\mathcal{R}_e$ from our loss function and show performance as "NoE". Performance of "NoE" indicates that our solution can still work well compared with competing methods as shown in Table 7.1, while the edge-preserved regularization can further boost our performance.

4) Hard-mask performance with regrading to λ in Eq. 7.7: We define λ as a threshold to distinguish inliers and ourliers, and set it to a small number (thus around 5% pixels will be selected as inliers in each iteration) in our experiments. To test how the network performs with different λ, we carried out experiments with λ in a range of $[0, 0.05]$[7] and show performance in Fig. 7.6. We observe relatively stable performance with small λ settings.

5) Soft-mask performance with regrading to k in Eq. 7.8: k is used to control the descent degree of the soft weights. We design extra experiments with respect to k in the range of $[1, 20]$[8], and show the performance as Fig. 7.7. Similar to Fig. 7.6, we again observe relatively stable performance with k in the range of $[1, 20]$, which further illustrate the robutstness of our solution.

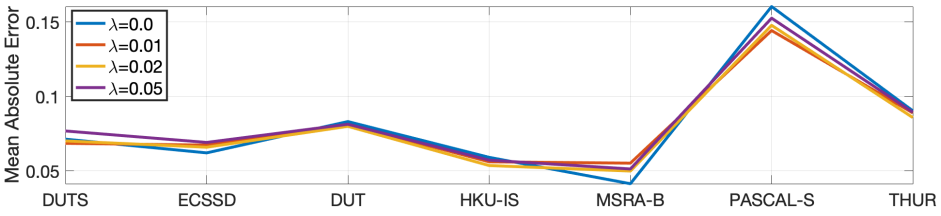


Figure 7.6: Model performance with regard to different λ in Eq. 7.7.

6) Learn from clean data: To test how our solution performs with clean labels available, we conducted two experiments: 1) training directly with clean label ("Clean"); 2) performing our solution by treating clean labels as noisy ("C_O"). The gap between "Clean" and "C_O" suggests the importance of noise handling even when the clean labelling is provided. Furthermore, we notice that the identified noise in "C_O" mainly along object edges, which is also consistent with the observation in [254] that

[6]Both of them are dilated version following DeeplabV2 [29]

[7]We define a smaller range for λ to avoid selecting too many outliers.

[8]Similar to hard mask strategy, we control k in a relatively compact range to avoid outliers contributing too much to network learning.

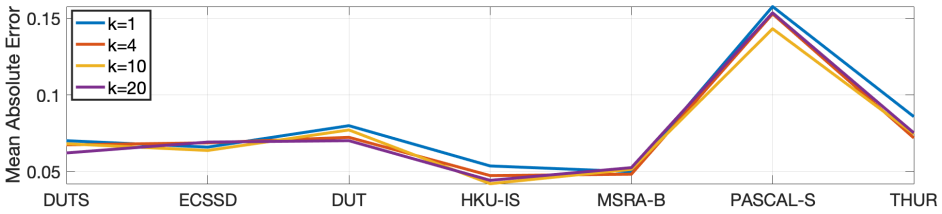


Figure 7.7: Model performance with regard to different k in Eq. 7.8.

the human annotated dense ground truth may not well align with object edges.
7) Learn with state-of-the-art saliency network? Our network is based on DeeplabV2 [29]. To test how our network performs with better network structure used, we implement our solution with BASNet [33] structure, and show the performance as "BAS_O". We can draw two conclusions from "BAS_O": 1) better backbone structure can further boost our performance; 2) comparing "BAS_O" with "C_O", we notice limited performance difference, which shows that the noise handling strategy is much more important than network structure in noisy labeling setting.
8) Learn from Web data: As our method does not rely on pixel-wise accurate labeling, we used the Webvision1.0 dataset [9] to train our models from *scratch* ("Web"). We notice comparable performance of "Web" with competing weakly/un-supervised methods, which indicate the effectiveness of our solution. Further, we observe a gap between our performance "Ours" in Table 7.2 and "Web". One reason is that we randomly initialized the weights in "Web", while for "Ours", we use the weights from Imagenet for weight initialization. Secondly, more complex images in Webvision1.0 dataset leading to more noisy training dataset than the benchmark DUTS[41] based noisy dataset. Note that, the "Web" experiment is totally unsupervised and self-explanatory, which further indicates the effectiveness of our proposed solution.

7.6 Conclusion

In this chaper, similar to Chapter 6, we focus on tackling a challenging and practical problem in saliency prediction (salient object detection) of learning saliency from a single noisy labelling. We formulated the problem as jointly learning a saliency prediction model and identifying the inliers. We proposed two approaches to address the problem, namely, hard mask selection and soft mask reweighting. Extensive experiments on benchmarking datasets demonstrate the superiority of our methods. One possible extension of our solution is to the multiple object segmentation task, e.g., semantic segmentation [29, 138], where we have access to noisy labeling of each semantic category. A straight forward solution can be achieved by defining the multi-category problem as multiple binary segmentation problem, where we simultaneously identify the noisy label for each category by directly applying the proposed solution. Another extension is combining the proposed solution with our latent variable techniques [60, 126, 65]. In this work, we iteratively identify clean

[9] https://www.vision.ee.ethz.ch/webvision/2017/index.html

labels with the variance of model prediction, assuming that the clean labels lead to relative consistent predictions across the epochs of learning. With the latent variable model based noise estimation module as in [65], we can define consistency loss between two types of noise, leading to more accurate and reliable noise estimation.

Chapter 8

Conclusion

Saliency detection is a category-free technique, which intends to detect and segment the most informative regions of the image that attract human attention. The large amount of pixel-wise labeled dataset and the booming development of deep learning makes saliency detection a very active topic recently. However, this problem is yet to be solved, and new challenges still remains.

8.1 Summary and Contribution

In this thesis, we mainly focus on two main important issues about saliency detection: 1) uncertainty-aware saliency detection to model the subjective nature of saliency and 2) weakly-/un-supervised saliency detection techniques to learn saliency from easy-to-access labels. We then summarize our contributions, and compare the proposed approaches with future research directions.

8.1.1 Our Contributions

CVAE based Uncertainty-aware RGB-D Saliency Detection. Existing RGB-D saliency detection models design deterministic networks to produce a single prediction for each input image. We argue that such one-to-one mappings fail to model the uncertainty of labeling, representing the subjective nature of saliency. In this way, we propose the first generative model based RGB-D saliency detection network [60, 61] with a latent variable to model the labeling variants. In our framework, the latent variable is conditional on the input image, which captures the hidden attributes of the image, making our model a probabilistic network to produce distribution estimation instead of point estimation.
Complementary Learning for RGB-D Saliency Detection via Latent Variable Model. One main drawback of existing RGB-D saliency detection models is that they fail to explicitly model the complementary information of the RGB image and the depth data, where the former captures appearance information and the latter encodes geometry information. Existing methods fuse the two different modes feature-wise to achieve implicitly complementary learning. We introduce the first cross-mode fusion model [126] to explicitly model the complementary information for RGB-D

saliency detection. Particularly, we intend to minimize the mutual information of the two modes, where the mutual information can be approximated with the symmetric KL-divergence, making it easy to be implemented. Furthermore, we introduce the largest RGB-D saliency detection dataset (10x' larger than existing dataset) to boost this community.

Integrated Latent Variable Model and Energy-based Model for RGB Saliency Detection. The latent variable model introduces latent variables to the network, thus it can model the uncertainty of labeling. The energy-based model defines a unnormalized scalar (the energy) to measure the compatibility of the variants. We design a probabilistic coarse-to-fine learning framework [62] for RGB saliency detection, where the latent variable model produces initial prediction, and the energy-based model serves as higher-order similarity measure to measure the accuracy of the initial prediction and refine it further with the gradient-based MCMC.

Learning RGB Saliency with Scribble Annotation. Compared with pixel-wise labeling, scribble annotation is easy to obtain, which only takes a few seconds to label one image. We then introduce a new weakly supervised learning setting: learning saliency from scribble annotation [63]. We first relabel existing RGB saliency detection dataset with scribble, then we design an edge-aware network to learn from partial supervision (the scribble annotation) and recover the missing structure information with an auxiliary edge detection branch and a structure-aware loss function.

Learning RGB Saliency from Noisy Labeling. To further relieve the burden of labeling, we present an approach to learn saliency from noisy labeling [66], which can be easily obtained by using handcrafted feature based conventional methods. Within this setting, we design two different solutions: 1) noisy auto-encoder based [65] and 2) robust model fitting based [64]. For the former, we introduce an extra noise generator to approximate noise in the noisy labeling, and for the latter, we iteratively sample from the noisy labeling to select inlier (clean label) for model training.

8.1.2 Proposed Approaches Comparison and Extension

Latent variable model + mutual information regularizer: In [60, 61] (Chapter 2), we achieve RGB-D saliency detection via early fusion, where the RGB image and depth are concatenated in the input layer, thus we can focus on the generative learning pipeline of the latent variable model. In [126] (Chapter 3), the mutual information regularizer is proven effective for multi-modal learning. It is then straightforward to borrow the benefits of both strategies with a cross-level feature generative model that aims to achieve both effective stochastic predictions and effective multi-modal learning.

Latent variable model + energy-based refinement: In [62] (Chapter 4), we adopt a Langevin dynamic [264] based latent variable model, namely alternating back-propagation (ABP) [139], to produce an initial prediction, which is then refined with the energy-based model (EBM). As our focus is to analyse how the EBM can be used as fine predictor, we choose ABP [139] as the coarse generative saliency predictor

due to it's efficiency. A potential extension is to define the conditional variational auto-encoder (CVAE) [51] (in Chapter 2) as the latent variable model and evaluate how EBM can contribute within a CVAE-EBM framework.

Weakly supervised learning + latent variable model: We present an auxiliary edge detection module and a structure-aware loss function within our weakly supervised learning pipeline to recover the missing structure information with the scribble annotation in Chapter 5. One drawback of the solution is that the unlabeled pixels, especially the unlabeled pixels within the semantic instances, fail to contribute enough to the model updating. An useful extension is through the use of a latent variable model, where a latent variable can be learned to approximate the latent distribution of the pixel-wise annotation. The basic assumption is that the partial annotation (scribble) should lie in the same latent space as the full annotation (pixel-wise annotation).

Generative noise estimation + sampling based inlier/outlier discovery: In both [65] (Chapter 6) and [64] (Chapter 7), we study the learning from noisy labeling problem. The main drawback of the above techniques is the error propagation problem due to the less accurate noise modeling. To solve it, we can define the noise modeling based consistency loss, which aims to estimate the accuracy of the noise estimation module with noise from both the generative noise estimation module (Chapter 6) and the outlier discovery solution (Chapter 7).

8.2 Work Extension

As a basic computer vision task, saliency detection belongs to the "Dense prediction" task as shown in Fig. 8.1. We divide the existing dense prediction tasks into a classification task and a regression task. The former produces pixel-wise classification maps, which includes semantic segmentation [29], instance segmentation [250], saliency detection [42], *etc.* The latter regresses the pixel-wise prediction for the input image, including depth estimation [265], image super-resolution [266], image deblurring [267], *etc.* We claim that, although our solutions above are performed on saliency detection, the general ideas can be easily extended to other dense prediction tasks. The proposed generative model frameworks can be used in both classification tasks and regression tasks, as shown in Fig. 8.1, to achieve uncertainty estimation, representing model confidence of the predictions. The sampling based inlier and outlier identifying solution can be used in the depth estimation task to deal with the less accurate (or noisy) depth issue. Our scribble annotation based solution provides an alternative strategy for weakly-supervised semantic segmentation and instance segmentation.

8.2.1 Uncertainty-aware Semantic/Instance Segmentation

As discussed in [254], there exists a significant level of semantic boundary noise in existing semantic segmentation or instance segmentation ground truth maps. This mainly comes from the difficulty in labeling the precise boundaries of objects. The

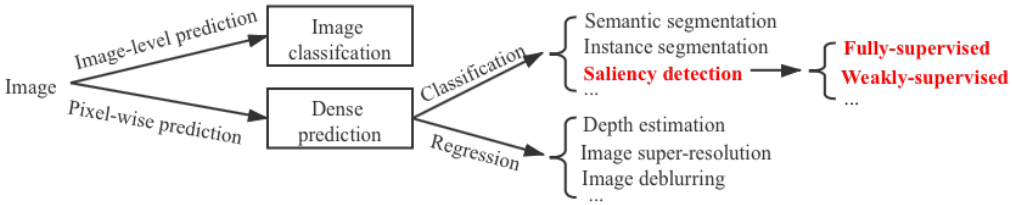


Figure 8.1: The widely studied computer vision tasks.

main issue of training directly with above less accurate ground truth map is that the trained model may over-fit on the noisy label, leading to poor generalization ability. Following our proposed solution in [65], a generative model can be trained for noise estimation, which is capable of capturing the noise in any distribution. Further, our uncertainty-aware solutions in [60, 61, 62] can also be adopted to model the aleatoric[1] uncertainty [192] of the less accurate dataset.

8.2.2 Depth Calibration

Depth data can be captured using a Kinect or computed directly from the stereo image pairs. For both scenarios, the generated depth may not be very accurate. For monocular depth estimation, current techniques simply train models to achieve mapping from an RGB image to the corresponding depth map. With our sampling based strategy in [64], we can iteratively identify pixels with noisy depth. As shown in [64], our sampling strategy can effectively select candidate clean depth as supervision. Further, for RGB-D based dense prediction tasks, our generative model pipeline in [60, 61] can be adopted to estimate the uncertainty of the depth map, achieving depth calibration for RGB-D image pair based dense prediction tasks.

8.2.3 Difficulty-aware Image Super-resolution/Deblurring

The purpose of image super-resolution or image deblurring is to generate high-resolution images from the low-quality input images. To make an image clear and high in quality, one can focus more on the semantic boundaries of the image, as an image of sharp boundaries usually has higher resolution than the images with blurring boundaries. With the proposed generative model in [60, 61, 62], the produced uncertainty map can serve as a guidance for hard pixel (which is usually the pixels along object boundaries) recognition. Based on the generated uncertainty map, more focus can be put on recovering the resolution of hard pixels for image super-resolution or image delurring.

[1] Aleatoric uncertainty captures noise inherent in the observations [192].

8.3 Future Work

In this thesis, we introduce generative model based RGB-D and RGB saliency detection networks. Further, we present our weakly supervised learning framework and learning from noisy labeling strategies. Although our solutions can help development of saliency detection to some extend, there still exists challenges to be taken in the future.

8.3.1 Attribute-aware Latent Variable Model for Saliency Detection

In our current generative model based frameworks, the latent variable is conditioned on the input image, while we cannot control each dimension of the latent variable. In other words, the dimension of the latent variable is tuned to achieve reasonable prediction without explaining meaning of each dimension. The main reason is that the current saliency datasets have one single ground truth for each image (or image pair for the RGB-D saliency dataset). In this situation, our loss function is simply defined between model prediction and the given one version of ground truth. To make the latent variable meaningful, an attribute-aware latent variable model can be investigated with multiple ground truth saliency maps for each input image.

8.3.2 Data Augmentation for Robust Model Training

The current saliency detection datasets [41, 42] are mainly obtained by relabeling existing datasets for the task of saliency detection, and it happens that the training dataset may be biased, which may lead to poor generalization ability during testing. Data augmentation is an effective solution to improve model generalization ability by augment the training dataset with new samples out of the distribution of current dataset. An effective data augmentation strategy can be used in both dataset preparing and model training, which has potential to further boost the performance of saliency detection, leading to better model generalization ability.

References

1. Y. Piao, W. Ji, J. Li, M. Zhang, and H. Lu, "Depth-induced multi-scale recurrent attention network for saliency detection," in *Int. Conf. Comput. Vis.*, 2019. (cited on pages xix, 2, 10, 13, 14, 16, 29, 30, 42, 43, 44, 47, 50, 54, 55, 56, and 57)

2. L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE T. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, 1998. (cited on pages 1 and 13)

3. J. K. Tsotsos, S. M. Culhane, W. K. Wai, Y. Lai, N. Davis, and F. Nuflo, "Modeling visual attention via selective tuning," *Artif. Intell.*, vol. 78, pp. 507–545, 1995. (cited on page 1)

4. B. Olshausen, C. Anderson, and D. Van Essen, "A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information," *Journal of Neuroscience*, vol. 13, no. 11, pp. 4700–4719, 1993. (cited on page 1)

5. C. Koch and S. Ullman, "Shifts in selective visual attention: towards the underlying neural circuitry," *Human neurobiology*, vol. 44, pp. 219–27, 1985. (cited on page 1)

6. D. Robinson and S. Petersen, "The pulvinar and visual salience," *Trends in Neurosciences*, vol. 15, no. 4, pp. 127–132, 1992. (cited on page 1)

7. A. M. Treisman and G. Gelade, "A feature-integration theory of attention," *Cognitive Psychology*, vol. 12, no. 1, pp. 97–136, 1980. (cited on page 1)

8. X. Huang, C. Shen, X. Boix, and Q. Zhao, "Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks," in *Int. Conf. Comput. Vis.*, Dec 2015, pp. 262–270. (cited on page 1)

9. T. Liu, J. Sun, N.-N. Zheng, X. Tang, and H.-Y. Shum, "Learning to detect a salient object," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2007, pp. 1–8. (cited on pages 1 and 117)

10. R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2009, pp. 1597–1604. (cited on page 1)

11. L. Marchesotti, C. Cifarelli, and G. Csurka, "A framework for visual saliency detection with applications to image thumbnailing," in *Int. Conf. Comput. Vis.*, 2009, pp. 2232–2239. (cited on page 1)

12. G.-X. Zhang, M.-M. Cheng, S.-M. Hu, and R. R. Martin, "A shape-preserving approach to image resizing," *Computer Graphics Forum*, vol. 28, no. 7, pp. 1897–1906, 2009. (cited on page 1)

13. V. Navalpakkam and L. Itti, "An integrated model of top-down and bottom-up attention for optimizing detection speed," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2006, pp. 2049–2056. (cited on page 1)

14. J. Li, R. Ma, and J. Ding, "Saliency-seeded region merging: Automatic object segmentation," in *Asian Conf. Pattern Recog.*, Nov 2011, pp. 691–695. (cited on page 1)

15. S. Sharma, R. Kiros, and R. Salakhutdinov, "Action recognition using visual attention," in *NIPS Time Series Workshop*, 2015. (cited on page 1)

16. K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *Int. Conf. Mach. Learn.*, vol. 37, 2015, pp. 2048–2057. (cited on page 1)

17. S. J. Oh, R. Benenson, A. Khoreva, Z. Akata, M. Fritz, and B. Schiele, "Exploiting saliency for object segmentation from image level labels," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 4410–4419. (cited on page 1)

18. H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, "Salient object detection: A discriminative regional feature integration approach," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 2083–2090. (cited on pages 2 and 118)

19. H.-H. Yeh, K.-H. Liu, and C.-S. Chen, "Salient object detection via local saliency estimation and global homogeneity refinement," *Pattern Recognition*, vol. 47, no. 4, pp. 1740 – 1750, 2014. (cited on page 2)

20. P. Jiang, H. Ling, J. Yu, and J. Peng, "Salient region detection by UFO: Uniqueness, focusness and objectness," in *Int. Conf. Comput. Vis.*, 2013, pp. 1976–1983. (cited on page 2)

21. M.-M. Cheng, J. Warrell, W.-Y. Lin, S. Zheng, V. Vineet, and N. Crook, "Efficient salient region detection with soft image abstraction," in *Int. Conf. Comput. Vis.*, 2013, pp. 1529–1536. (cited on page 2)

22. S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," *IEEE T. Pattern Anal. Mach. Intell.*, vol. 34, no. 10, pp. 1915–1926, 2012. (cited on page 2)

23. J. Kim, D. Han, Y.-W. Tai, and J. Kim, "Salient region detection via high-dimensional color transform," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 883–890. (cited on pages 2 and 75)

24. W. Zhu, S. Liang, Y. Wei, and J. Sun, "Saliency optimization from robust background detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 2814–2821. (cited on pages 2, 9, 31, 75, 92, 99, 103, 107, 108, 117, 118, and 120)

25. L. Huo, L. Jiao, S. Wang, and S. Yang, "Object-level saliency detection with color attributes," *Pattern Recognition*, vol. 49, pp. 162 – 173, 2016. (cited on page 2)

26. C. Yang, L. Zhang, H. Lu, X. Ruan, and M. Yang, "Saliency detection via graph-based manifold ranking," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 3166–3173. (cited on pages 2, 31, 38, 70, 86, 92, 99, 103, 117, and 120)

27. A. Borji, M. Cheng, Q. Hou, H. Jiang, and J. Li, "Salient object detection: A survey," *Computational Visual Media*, vol. 5, pp. 117–150, 2019. (cited on pages 2, 107, 108, and 111)

28. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 770–778. (cited on pages 2, 4, 19, 23, 29, 33, 45, 54, 69, 70, 99, 105, and 117)

29. L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE T. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, 2017. (cited on pages 2, 6, 7, 74, 76, 78, 79, 83, 117, 121, 122, and 127)

30. G. Li and Y. Yu, "Visual saliency based on multiscale deep features," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 5455–5463. (cited on pages 2, 38, 70, 86, 99, and 117)

31. N. Liu, J. Han, and M.-H. Yang, "Picanet: Learning pixel-wise contextual attention for saliency detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 3089–3098. (cited on pages 2, 31, 37, 63, 70, 86, 92, and 111)

32. Z. Luo, A. Mishra, A. Achkar, J. Eichel, S. Li, and P.-M. Jodoin, "Non-local deep features for salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 6609–6617. (cited on pages 2, 27, 31, 37, 63, 70, 73, 84, 86, 100, 101, 111, and 118)

33. X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, and M. Jagersand, "Basnet: Boundary-aware salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. (cited on pages 2, 37, 42, 61, 63, 69, 70, 73, 76, 78, 86, 92, 94, 100, 103, 107, 111, 118, and 122)

34. S. W. Jun Wei and Q. Huang, "F3net: Fusion, feedback and focus for salient object detection," in *AAAI Conf. Art. Intell.*, 2020. (cited on pages 2, 13, 27, 31, 36, 42, 61, 63, 69, 70, 92, 94, 107, 111, 118, and 119)

35. W. Wang, J. Shen, M.-M. Cheng, and L. Shao, "An iterative and cooperative top-down and bottom-up inference network for salient object detection," in *IEEE*

Conf. Comput. Vis. Pattern Recog., 2019. (cited on pages 2, 13, 42, 61, 63, 78, 92, and 94)

36. P. Zhang, D. Wang, H. Lu, H. Wang, and X. Ruan, "Amulet: Aggregating multi-level convolutional features for salient object detection," in *Int. Conf. Comput. Vis.*, 2017. (cited on pages 2 and 92)

37. Z. Wu, L. Su, and Q. Huang, "Cascaded partial decoder for fast and accurate salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. (cited on pages 2, 31, 61, 63, 69, 86, 92, 94, and 100)

38. X. Zhang, T. Wang, J. Qi, H. Lu, and G. Wang, "Progressive attention guided recurrent network for salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 714–722. (cited on pages 2, 86, and 92)

39. R. Wu, M. Feng, W. Guan, D. Wang, H. Lu, and E. Ding, "A mutual learning method for salient object detection with intertwined multi-supervision," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. (cited on pages 2, 37, 70, 86, 92, 100, and 118)

40. M. Feng, H. Lu, and E. Ding, "Attentive feedback network for boundary-aware salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. (cited on pages 2, 31, 37, 70, 86, 92, 100, and 118)

41. L. Wang, H. Lu, Y. Wang, M. Feng, D. Wang, B. Yin, and X. Ruan, "Learning to detect salient objects with image-level supervision," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 136–145. (cited on pages 2, 4, 6, 7, 38, 43, 48, 59, 63, 70, 76, 77, 78, 86, 92, 94, 99, 100, 107, 111, 117, 118, 122, and 129)

42. M. Cheng, G. Zhang, N. Mitra, X. Huang, and S.-M. Hu, "Global contrast based salient region detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 409–416. (cited on pages 2, 92, 104, 107, 127, and 129)

43. D.-P. Fan, Z. Lin, Z. Zhang, M. Zhu, and M.-M. Cheng, "Rethinking RGB-D salient object detection: Models, datasets, and large-scale benchmarks," *IEEE T. Neural Netw. Learn. Syst.*, 2020. (cited on pages 2, 14, 25, 28, 30, 31, 42, 43, 44, 54, 57, and 59)

44. H. Zhou, X. Xie, J.-H. Lai, Z. Chen, and L. Yang, "Interactive two-stream decoder for accurate and fast saliency detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2020. (cited on pages 2 and 70)

45. H. Chen and Y. Li, "Progressively complementarity-aware fusion network for RGB-D salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 3051–3060. (cited on pages 2, 16, 29, 33, 42, and 44)

46. J.-J. Liu, Q. Hou, M.-M. Cheng, J. Feng, and J. Jiang, "A simple pooling-based design for real-time salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. (cited on pages 2, 61, 63, 70, 73, 107, 111, and 118)

47. K. Fu, D.-P. Fan, G.-P. Ji, and Q. Zhao, "Jl-dcf: Joint learning and densely-cooperative fusion framework for rgb-d salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. (cited on pages 2, 13, 14, 16, 17, 55, 56, and 57)

48. H. Chen and Y. Li, "Three-stream attention-aware network for RGB-D salient object detection," *IEEE T. Image Process.*, pp. 2825–2835, 2019. (cited on pages 2, 14, 16, 29, 42, 44, and 55)

49. J. Han, H. Chen, N. Liu, C. Yan, and X. Li, "CNNs-based RGB-D saliency detection via cross-view transfer and multiview fusion," *IEEE T. Cybern.*, pp. 3171–3183, 2018. (cited on pages 2, 14, 16, 29, 33, 37, 42, 44, and 55)

50. D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Int. Conf. Learn. Represent.*, 2013. (cited on pages 3, 4, 5, 16, 17, 20, 45, 46, 63, 69, and 93)

51. K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Adv. Neural Inform. Process. Syst.*, 2015, pp. 3483–3491. (cited on pages 3, 4, 5, 9, 15, 16, 17, 18, 20, 21, 22, 24, 64, 69, 71, 104, and 127)

52. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Adv. Neural Inform. Process. Syst.*, 2014, pp. 2672–2680. (cited on pages 3, 4, 5, 16, 17, and 45)

53. D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for boltzmann machines," *Cognitive Science*, vol. 9, no. 1, pp. 147 – 169, 1985. (cited on page 4)

54. G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Comput.*, vol. 14, no. 8, p. 1771–1800, 2002. (cited on page 4)

55. R. Salakhutdinov and G. Hinton, "Deep boltzmann machines," *Proceedings of AISTATS 2009*, vol. 5, pp. 448–455, 01 2009. (cited on pages 4 and 6)

56. J. Xie, Y. Lu, R. Gao, S.-C. Zhu, and Y. N. Wu, "Cooperative training of descriptor and generator networks," *IEEE T. Pattern Anal. Mach. Intell.*, 2018. (cited on pages 4, 45, 62, 63, and 64)

57. Y. LeCun, S. Chopra, R. Hadsell, F. J. Huang, and et al., "A tutorial on energy-based learning," in *PREDICTING STRUCTURED DATA*. MIT Press, 2006. (cited on pages 4, 6, and 63)

58. Y. Du and I. Mordatch, "Implicit generation and modeling with energy based models," in *Adv. Neural Inform. Process. Syst.*, 2019, pp. 3608–3618. (cited on pages 4, 6, and 63)

59. J. Xie, Y. Lu, S.-C. Zhu, and Y. Wu, "A theory of generative convnet," in *Int. Conf. Mach. Learn.*, ser. Proceedings of Machine Learning Research, vol. 48, 2016, pp. 2635–2644. (cited on pages 4, 6, and 63)

60. J. Zhang, D.-P. Fan, Y. Dai, S. Anwar, F. Sadat Saleh, T. Zhang, and N. Barnes, "Uc-net: Uncertainty inspired rgb-d saliency detection via conditional variational autoencoders," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. (cited on pages 4, 6, 13, 15, 16, 18, 29, 30, 41, 42, 44, 45, 55, 56, 57, 60, 63, 64, 75, 78, 92, 122, 125, 126, and 128)

61. J. Zhang, D.-P. Fan, Y. Dai, S. Anwar, F. Saleh, S. Aliakbarian, and N. Barnes, "Uncertainty inspired rgb-d saliency detection," *IEEE T. Pattern Anal. Mach. Intell.*, 2021. (cited on pages 4, 6, 125, 126, and 128)

62. J. Zhang, J. Xie, Z. Zheng, and N. Barnes, "Probabilistic coarse-to-fine saliency prediction," in *CVPR2021 under review*, 2021. (cited on pages 4, 10, 126, and 128)

63. J. Zhang, X. Yu, A. Li, P. Song, B. Liu, and Y. Dai, "Weakly-supervised salient object detection via scribble annotations," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. (cited on pages 4, 6, 8, 10, 13, 59, 62, 63, 67, 70, 72, 73, 92, 94, 111, and 126)

64. J. Zhang, Y. Dai, T. Zhang, M. Harandi, N. Barnes, and R. Hartley, "Learning saliency from single noisy labelling: A robust model fitting perspective," *IEEE T. Pattern Anal. Mach. Intell.*, 2020. (cited on pages 4, 9, 11, 126, 127, and 128)

65. J. Zhang, J. Xie, and N. Barnes, "Learning noise-aware encoder-decoder from noisy labels by alternating back-propagation for saliency detection," in *Eur. Conf. Comput. Vis.*, 2020. (cited on pages 4, 6, 9, 11, 63, 122, 123, 126, 127, and 128)

66. J. Zhang, T. Zhang, Y. Dai, M. Harandi, and R. Hartley, "Deep unsupervised saliency detection: A multiple noisy labeling perspective," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 9029–9038. (cited on pages 4, 9, 13, 63, 76, 78, 79, 86, 92, 93, 94, 100, 105, 108, 111, 112, 118, 119, and 126)

67. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Int. Conf. Learn. Represent.*, 2014. (cited on pages 4, 33, 81, 84, 85, 97, 99, 100, 105, and 121)

68. M. Mirza and S. Osindero, "Conditional generative adversarial nets," *CoRR*, vol. abs/1411.1784, 2014. (cited on pages 5, 16, 17, 64, 69, and 71)

69. B. Li, Z. Sun, and Y. Guo, "Supervae: Superpixelwise variational autoencoder for salient object detection," in *AAAI Conf. Art. Intell.*, 2019, pp. 8569–8576. (cited on pages 5, 17, and 45)

70. I. Gulrajani, K. Kumar, F. Ahmed, A. A. Taiga, F. Visin, D. Vazquez, and A. Courville, "Pixelvae: A latent variable model for natural images," in *Int. Conf. Learn. Represent.*, 2016. (cited on pages 5 and 17)

71. X. Yan, A. Rastogi, R. Villegas, K. Sunkavalli, E. Shechtman, S. Hadap, E. Yumer, and H. Lee, "Mt-vae: Learning motion transformations to generate multimodal human dynamics," in *Eur. Conf. Comput. Vis.*, 2018, pp. 276–293. (cited on pages 5 and 17)

72. C. F. Baumgartner, K. C. Tezcan, K. Chaitanya, A. M. Hötker, U. J. Muehlematter, K. Schawkat, A. S. Becker, O. Donati, and E. Konukoglu, "Phiseg: Capturing uncertainty in medical image segmentation," in *MICCAI*, 2019, pp. 119–127. (cited on pages 5 and 17)

73. S. Kohl, B. Romera-Paredes, C. Meyer, J. De Fauw, J. R. Ledsam, K. Maier-Hein, S. M. A. Eslami, D. Jimenez Rezende, and O. Ronneberger, "A probabilistic u-net for segmentation of ambiguous images," in *Adv. Neural Inform. Process. Syst.*, 2018, pp. 6965–6975. (cited on pages 5, 17, and 64)

74. A. Abid and J. Y. Zou, "Contrastive variational autoencoder enhances salient features," *CoRR*, vol. abs/1902.04601, 2019. (cited on pages 6 and 17)

75. J. Walker, C. Doersch, H. Mulam, and M. Hebert, "An uncertain future: Forecasting from static images using variational autoencoders," in *Eur. Conf. Comput. Vis. Worksh.*, 2016, pp. 835–851. (cited on pages 6 and 17)

76. S. Aliakbarian, F. S. Saleh, M. Salzmann, L. Petersson, and S. Gould, "A stochastic conditioning scheme for diverse human motion prediction," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. (cited on pages 6, 17, and 35)

77. P. Esser, E. Sutter, and B. Ommer, "A variational u-net for conditional appearance and shape generation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 8857–8865. (cited on pages 6 and 17)

78. Q. Tan, L. Gao, Y.-K. Lai, and S. Xia, "Variational autoencoders for deforming 3d mesh models," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. (cited on pages 6 and 17)

79. L. Yi, W. Zhao, H. Wang, M. Sung, and L. J. Guibas, "Gspn: Generative shape proposal network for 3d instance segmentation in point cloud," in *Eur. Conf. Comput. Vis.*, 2019. (cited on pages 6, 17, and 95)

80. P. Luc, C. Couprie, S. Chintala, and J. Verbeek, "Semantic segmentation using adversarial networks," in *Adv. Neural Inform. Process. Syst. Worksh.*, 2016. (cited on pages 6, 17, and 64)

81. X. Zhang, X. Zhu, . X. Zhang, N. Zhang, P. Li, and L. Wang, "Seggan: Semantic segmentation with generative adversarial network," in *2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM)*, 2018, pp. 1–5. (cited on pages 6, 17, and 64)

82. Y. Xue, T. Xu, H. Zhang, R. Long, and X. Huang, "Segan: Adversarial network with multi-scale l_1 loss for medical image segmentation," *Neuroinformatics*, vol. 16, 06 2017. (cited on pages 6, 17, and 64)

83. J. Pan, C. Canton, K. McGuinness, N. E. O'Connor, J. Torres, E. Sayrol, and X. a. Giro-i Nieto, "Salgan: Visual saliency prediction with generative adversarial networks," in *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, 2017. (cited on pages 6, 17, 45, and 64)

84. H. Yu and X. Cai, "Saliency detection by conditional generative adversarial network," in *Ninth International Conference on Graphic and Image Processing*, 04 2018, p. 253. (cited on pages 6, 17, and 64)

85. W.-C. Hung, Y.-H. Tsai, Y.-T. Liou, Y.-Y. Lin, and M.-H. Yang, "Adversarial learning for semi-supervised semantic segmentation," in *Brit. Mach. Vis. Conf.*, 2018. (cited on pages 6, 17, 64, 70, and 79)

86. N. Souly, C. Spampinato, and M. Shah, "Semi supervised semantic segmentation using generative adversarial network," in *Int. Conf. Comput. Vis.*, 2017, pp. 5689–5697. (cited on pages 6, 17, and 64)

87. J. Ngiam, Z. Chen, P. Koh, and A. Ng, "Learning deep energy models," in *Int. Conf. Mach. Learn.*, 01 2011, pp. 1105–1112. (cited on pages 6 and 63)

88. T. Kim and Y. Bengio, "Deep directed generative models with energy-based probability estimation," *CoRR*, vol. abs/1606.03439, 2016. (cited on pages 6 and 63)

89. J. J. Zhao, M. Mathieu, and Y. LeCun, "Energy-based generative adversarial network," *CoRR*, vol. abs/1609.03126, 2016. (cited on pages 6 and 63)

90. R. Gao, Y. Lu, J. Zhou, S. Zhu, and Y. Wu, "Learning generative convnets via multi-grid modeling and sampling," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 06 2018, pp. 9155–9164. (cited on pages 6 and 63)

91. R. Kumar, A. Goyal, A. C. Courville, and Y. Bengio, "Maximum entropy generators for energy-based models," *CoRR*, vol. abs/1901.08508, 2019. (cited on pages 6 and 63)

92. E. Nijkamp, M. Hill, S.-C. Zhu, and Y. N. Wu, "Learning non-convergent non-persistent short-run mcmc toward energy-based model," in *Adv. Neural Inform. Process. Syst.*, 2019. (cited on pages 6 and 63)

93. C. Finn, P. F. Christiano, P. Abbeel, and S. Levine, "A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models," in *Adv. Neural Inform. Process. Syst. Worksh.*, 2016. (cited on pages 6 and 63)

94. A. G. Alias Parth Goyal, N. R. Ke, S. Ganguli, and Y. Bengio, "Variational walk-back: Learning a transition operator as a stochastic recurrent net," in *Adv. Neural Inform. Process. Syst.*, 2017, pp. 4392–4402. (cited on pages 6 and 63)

95. W. Grathwohl, K.-C. Wang, J.-H. Jacobsen, D. Duvenaud, M. Norouzi, and K. Swersky, "Your classifier is secretly an energy based model and you should treat it like one," in *Int. Conf. Learn. Represent.*, 2020. (cited on pages 6 and 63)

96. G. Desjardins, Y. Bengio, and A. C. Courville, "On tracking the partition function," in *Adv. Neural Inform. Process. Syst.*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds., 2011, pp. 2501–2509. (cited on pages 6 and 63)

97. A. Mnih and G. Hinton, "Learning nonlinear constraints with contrastive back-propagation," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 2, 2005, pp. 1302–1307. (cited on page 6)

98. G. Li, Y. Xie, and L. Lin, "Weakly supervised salient object detection using image labels," in *AAAI Conf. Art. Intell.*, 2018. (cited on pages 6, 7, 63, 76, 78, 86, 92, 94, 100, 111, and 118)

99. D. T. Nguyen, M. Dax, C. K. Mummadi, T.-P.-N. Ngo, T. H. P. Nguyen, Z. Lou, and T. Brox, "Deepusps: Deep robust unsupervised saliency prediction with self-supervision," in *Adv. Neural Inform. Process. Syst.*, 2019. (cited on pages 6, 9, 63, 76, 78, 92, 93, 94, 105, 108, 111, and 112)

100. B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. (cited on page 6)

101. K.-J. Hsu, Y.-Y. Lin, and Y.-Y. Chuang, "Weakly supervised saliency detection with a category-driven map generator," in *Brit. Mach. Vis. Conf.*, 2017. (cited on pages 7 and 78)

102. Y. Zeng, Y. Zhuge, H. Lu, and L. Zhang, "Multi-source weak supervision for saliency detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 6074–6083. (cited on pages 7, 76, 78, 86, 94, and 100)

103. J. Dai, K. He, and J. Sun, "Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation," in *Int. Conf. Comput. Vis.*, 2015, pp. 1635–1643. (cited on pages 7 and 79)

104. A. Khoreva, R. Benenson, J. Hosang, M. Hein, and B. Schiele, "Simple does it: Weakly supervised instance and semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 876–885. (cited on pages 7 and 79)

105. Y. Wei, H. Xiao, H. Shi, Z. Jie, J. Feng, and T. S. Huang, "Revisiting dilated convolution: A simple approach for weakly- and semi- supervised semantic

segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 7268–7277. (cited on pages 7, 79, and 81)

106. X. Wang, S. You, X. Li, and H. Ma, "Weakly-supervised semantic segmentation by iteratively mining common object features," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 1354–1362. (cited on pages 7 and 79)

107. Z. Huang, X. Wang, J. Wang, W. Liu, and J. Wang, "Weakly-supervised semantic segmentation network with deep seeded region growing," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 7014–7023. (cited on pages 7 and 79)

108. P. Vernaza and M. Chandraker, "Learning random-walk label propagation for weakly-supervised semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 2953–2961. (cited on pages 7 and 79)

109. Y. Wang, Y. Yang, Z. Yang, L. Zhao, P. Wang, and W. Xu, "Occlusion aware unsupervised learning of optical flow," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. (cited on pages 7, 26, 27, 49, 50, 73, 82, 88, 93, and 97)

110. A. Kolesnikov and C. H. Lampert, "Seed, expand and constrain: Three principles for weakly-supervised image segmentation," in *Int. Conf. Comput. Vis.*, 2016, pp. 695–711. (cited on pages 7, 8, 79, and 82)

111. M. Tang, A. Djelouah, F. Perazzi, Y. Boykov, and C. Schroers, "Normalized cut loss for weakly-supervised cnn segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 1818–1827. (cited on pages 7, 76, 79, 80, 81, 113, and 115)

112. A. Obukhov, S. Georgoulis, D. Dai, and L. Van Gool, "Gated crf loss for weakly supervised semantic image segmentation," in *NeurIPS*, 2019. (cited on pages 8 and 79)

113. D.-H. Lee, "Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks," in *Int. Conf. Mach. Learn. Workshp*, 07 2013. (cited on pages 8 and 111)

114. J. Goldberger and E. Ben-Reuven, "Training deep neural networks using a noise adaptation layer," in *Int. Conf. Learn. Represent.*, 2017. (cited on pages 8, 95, and 111)

115. H.-S. Chang, E. Learned-Miller, and A. Mccallum, "Active bias: Training a more accurate neural network by emphasizing high variance samples," in *Adv. Neural Inform. Process. Syst.*, 2017. (cited on pages 8, 111, 114, and 115)

116. Z. Zhang and M. R. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *Adv. Neural Inform. Process. Syst.*, 2018. (cited on pages 8 and 115)

117. I. Jindal, M. Nokleby, and X. Chen, "Learning deep networks from noisy labels with dropout regularization," in *Int. Conf. Data Mining*, Dec 2016, pp. 967–972. (cited on page 8)

118. L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, "Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels," in *Int. Conf. Mach. Learn.*, 2018. (cited on pages 8, 95, 108, 111, and 113)

119. G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *Adv. Neural Inform. Process. Syst. Worksh.*, 2015. (cited on page 8)

120. T. Liu and D. Tao, "Classification with noisy labels by importance reweighting," *IEEE T. Pattern Anal. Mach. Intell.*, vol. 38, no. 3, pp. 447–461, 2016. (cited on pages 8, 95, and 111)

121. N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari, "Learning with noisy labels," in *Adv. Neural Inform. Process. Syst.*, 2013, pp. 1196–1204. (cited on page 8)

122. M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," in *Int. Conf. Mach. Learn.*, 2018. (cited on pages 8 and 111)

123. D. T. Nguyen, C. K. Mummadi, T. P. N. Ngo, T. H. P. Nguyen, L. Beggel, and T. Brox, "Self: Learning to filter noisy labels with self-ensembling," in *Int. Conf. Learn. Represent.*, 2020. (cited on pages 8, 109, 111, and 113)

124. D. Zhang, J. Han, and Y. Zhang, "Supervision by fusion: Towards unsupervised learning of deep salient object detector," in *Int. Conf. Comput. Vis.*, 2017, pp. 4068–4076. (cited on pages 9, 63, 78, 79, 86, 92, 93, 94, 105, 108, 111, 112, 117, and 118)

125. D.-P. Fan, Y. Zhai, A. Borji, J. Yang, and L. Shao, "BBS-Net: RGB-D Salient Object Detection with a Bifurcated Backbone Strategy Network," in *Eur. Conf. Comput. Vis.*, 2020. (cited on pages 10, 16, 42, 44, 55, and 57)

126. J. Zhang, D.-P. Fan, Y. Dai, X. Yu, Y. Zhong, N. Barnes, and L. Shao, "Rgb-d saliency detection via cascaded mutual information minimization," in *Int. Conf. Comput. Vis.*, 2021. (cited on pages 10, 122, 125, and 126)

127. D. Tanaka, D. Ikami, T. Yamasaki, and K. Aizawa, "Joint optimization framework for learning with noisy labels," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. (cited on pages 11, 108, and 112)

128. R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 1597–1604. (cited on page 13)

129. Y. Liu, Q. Zhang, D. Zhang, and J. Han, "Employing deep part-object relationships for salient object detection," in *Int. Conf. Comput. Vis.*, 2019. (cited on pages 13, 92, and 94)

130. Z. Wu, L. Su, and Q. Huang, "Stacked cross refinement network for edge-aware salient object detection," in *Int. Conf. Comput. Vis.*, 2019. (cited on pages 13, 31, 37, 42, 61, 63, 69, 70, 73, 75, 78, 100, 107, 111, and 118)

131. J.-X. Zhao, Y. Cao, D.-P. Fan, M.-M. Cheng, X.-Y. Li, and L. Zhang, "Contrast prior and fluid pyramid integration for rgbd salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. (cited on pages 13, 14, 16, 29, 30, 31, 33, 42, 44, 55, 56, 57, and 94)

132. M. Zhang, W. Ren, Y. Piao, Z. Rong, and H. Lu, "Select, supplement and focus for rgb-d saliency detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. (cited on pages 13, 16, 17, 42, and 44)

133. N. Liu, N. Zhang, and J. Han, "Learning selective self-mutual attention for rgb-d saliency detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. (cited on pages 13, 16, 42, 44, 55, 56, and 57)

134. Y. Piao, Z. Rong, M. Zhang, W. Ren, and H. Lu, "A2dele: Adaptive and attentive depth distiller for efficient rgb-d salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. (cited on pages 13, 16, 17, 42, 44, 55, 56, and 57)

135. O. Le Meur and T. Baccino, "Methods for comparing scanpaths and saliency maps: strengths and weaknesses," *Behavior Research Methods*, vol. 45, no. 1, pp. 251–266, 2013. (cited on page 14)

136. J. M. Henderson and T. R. Hayes, "Meaning-based guidance of attention in scenes as revealed by meaning maps," *Nature Human Behaviour*, vol. 1, no. 10, pp. 743–747, 2017. (cited on page 14)

137. L. Itti and C. Koch, "A saliency-based search mechanism for overt and covert shifts of visual attention," *Vis. Res.*, vol. 40, no. 10, pp. 1489 – 1506, 2000. (cited on page 14)

138. J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 3431–3440. (cited on pages 14, 19, 107, 108, and 122)

139. T. Han, Y. Lu, S. Zhu, and Y. Wu, "Alternating back-propagation for generator network," in *AAAI Conf. Art. Intell.*, 02 2017. (cited on pages 15, 16, 18, 20, 23, 24, 93, 105, and 126)

140. R. M. Neal, *MCMC Using Hamiltonian Dynamics*. CRC Press, 2011, vol. 54, pp. 113–162. (cited on pages 15, 16, 20, and 96)

141. J. He, D. Spokoyny, G. Neubig, and T. Berg-Kirkpatrick, "Lagging inference networks and posterior collapse in variational autoencoders," in *Int. Conf. Learn. Represent.*, 2019. (cited on pages 15, 18, 23, 24, 35, and 71)

142. C. K. Sø nderby, T. Raiko, L. Maalø e, S. r. K. Sø nderby, and O. Winther, "Ladder variational autoencoders," in *Adv. Neural Inform. Process. Syst.*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds., 2016, pp. 3738–3746. (cited on pages 15, 18, 21, and 24)

143. I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework," in *Int. Conf. Learn. Represent.*, 2017. (cited on pages 15, 18, 21, and 24)

144. J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 3146–3154. (cited on pages 15, 20, and 47)

145. L. Qu, S. He, J. Zhang, J. Tian, Y. Tang, and Q. Yang, "RGBD salient object detection via deep fusion," *IEEE T. Image Process.*, vol. 26, no. 5, pp. 2274–2285, 2017. (cited on pages 16, 29, 33, 42, and 44)

146. N. Wang and X. Gong, "Adaptive fusion for RGB-D salient object detection," *arXiv:1901.01369*, 2019. (cited on pages 16, 29, 30, 31, 33, 42, 44, and 55)

147. H. Chen, Y. Li, and D. Su, "Multi-modal fusion network with multi-scale multi-path and cross-modal interactions for RGB-D salient object detection," *Pattern Recognit.*, vol. 86, pp. 376–385, 2019. (cited on pages 16, 29, 33, 42, and 44)

148. K. Fu, D.-P. Fan, G.-P. Ji, Q. Zhao, J. Shen, and C. Zhu, "Siamese network for rgb-d salient object detection and beyond," *arXiv preprint arXiv:2008.12134*, 2020. (cited on page 16)

149. Y. Zhai, D.-P. Fan, J. Yang, A. Borji, L. Shao, J. Han, and L. Wang, "Bifurcated backbone strategy for rgb-d salient object detection," *arXiv e-prints*, pp. arXiv–2007, 2020. (cited on page 16)

150. W. Ji, J. Li, M. Zhang, Y. Piao, and H. Lu, "Accurate rgb-d salient object detection via collaborative learning," in *Eur. Conf. Comput. Vis.*, 2020. (cited on pages 16, 42, 44, 55, 56, and 57)

151. Y. Pang, L. Zhang, X. Zhao, and H. Lu, "Hierarchical dynamic filtering network for rgb-d salient object detection," in *Eur. Conf. Comput. Vis.*, 2020. (cited on pages 16, 42, 44, 55, and 56)

152. Z. Zhang, Z. Lin, J. Xu, W. Jin, S.-P. Lu, and D.-P. Fan, "Bilateral attention network for rgb-d salient object detection," *arXiv preprint arXiv:2004.14582*, 2020. (cited on pages 16, 42, 44, and 55)

153. T. Zhou, D.-P. Fan, M.-M. Cheng, J. Shen, and L. Shao, "RGB-D Salient Object Detection: A Survey," *arXiv preprint arXiv:2008.00230*, 2020. (cited on pages 17 and 111)

154. D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Int. Conf. Mach. Learn.*, 2014, pp. 1278–1286. (cited on pages 17 and 63)

155. S. Aliakbarian, F. S. Saleh, M. Salzmann, L. Petersson, and S. Gould, "Sampling good latent variables via cpp-vaes: Vaes with condition posterior as prior," *arXiv preprint arXiv:1912.08521*, 2019. (cited on page 17)

156. Y. Tang and X. Wu, "Salient object detection using cascaded convolutional neural networks and adversarial learning," *IEEE T. Multimedia*, vol. 21, no. 9, pp. 2237–2247, 2019. (cited on pages 17 and 18)

157. B. Jiang, Z. Zhou, X. Wang, and J. Tang, "cmsalgan: Rgb-d salient object detection with cross-view generative adversarial networks," *IEEE T. Multimedia*, 2019. (cited on pages 17 and 45)

158. P. Mukherjee, M. Sharma, M. Makwana, A. P. Singh, A. Upadhyay, A. Trivedi, B. Lall, and S. Chaudhury, "DSAL-GAN: denoising based saliency prediction with generative adversarial networks," *CoRR*, vol. abs/1904.01215, 2019. (cited on page 18)

159. P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," in *Adv. Neural Inform. Process. Syst.*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds., 2011, pp. 109–117. (cited on page 18)

160. S. Song, H. Yu, Z. Miao, J. Fang, K. Zheng, C. Ma, and S. Wang, "Multi-spectral salient object detection by adversarial domain adaptation," in *AAAI Conf. Art. Intell.*, 2020, pp. 12 023–12 030. (cited on page 18)

161. J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Int. Conf. Comput. Vis.*, 2017, pp. 2242–2251. (cited on page 18)

162. M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "Denseaspp for semantic segmentation in street scenes," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 3684–3692. (cited on pages 19, 20, 29, 47, and 81)

163. Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Eur. Conf. Comput. Vis.*, 2018. (cited on pages 19, 20, 68, 69, and 99)

164. A. Borji, M.-M. Cheng, H. Jiang, and J. Li, "Salient object detection: A benchmark," *IEEE T. Image Process.*, vol. 24, no. 12, pp. 5706–5722, 2015. (cited on pages 25 and 32)

165. C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 270–279. (cited on pages 26 and 82)

166. S. A. C. Yohanandan, A. G. Dyer, D. Tao, and A. Song, "Saliency preservation in low-resolution grayscale images," in *Eur. Conf. Comput. Vis.*, 2018. (cited on page 27)

167. R. Ju, L. Ge, W. Geng, T. Ren, and G. Wu, "Depth saliency based on anisotropic center-surround difference," in *IEEE Int. Conf. Image Process.*, 2014, pp. 1115–1119. (cited on pages 28, 29, 31, 33, 42, 43, 44, 47, 50, 54, 57, 58, and 59)

168. H. Peng, B. Li, W. Xiong, W. Hu, and R. Ji, "Rgbd salient object detection: a benchmark and algorithms," in *Eur. Conf. Comput. Vis.*, 2014, pp. 92–109. (cited on pages 28, 31, 33, 42, 43, 44, 47, 50, 54, 57, and 59)

169. Y. Niu, Y. Geng, X. Li, and F. Liu, "Leveraging stereopsis for saliency analysis," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 454–461. (cited on pages 28, 30, 31, 42, 44, 54, 57, 58, and 59)

170. N. Li, J. Ye, Y. Ji, H. Ling, and J. Yu, "Saliency detection on light field," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 2806–2813. (cited on pages 28, 31, 42, 44, 54, and 57)

171. Y. Cheng, H. Fu, X. Wei, J. Xiao, and X. Cao, "Depth enhanced saliency detection method," in *ACM ICIMCS*, 2014, pp. 23–27. (cited on pages 28, 31, 33, 42, 44, 54, and 57)

172. D.-P. Fan, M.-M. Cheng, Y. Liu, T. Li, and A. Borji, "Structure-measure: A new way to evaluate foreground maps," in *Int. Conf. Comput. Vis.*, 2017, pp. 4548–4557. (cited on pages 28, 54, 70, 84, 100, and 117)

173. D.-P. Fan, C. Gong, Y. Cao, B. Ren, M.-M. Cheng, and A. Borji, "Enhanced-alignment measure for binary foreground map evaluation," in *Int. Jt. Conf. Artif. Intell.*, 2018, pp. 698–704. (cited on pages 28, 29, 54, 55, 70, 84, 87, 100, and 117)

174. J. Ren, X. Gong, L. Yu, W. Zhou, and M. Ying Yang, "Exploiting global priors for rgb-d saliency detection," in *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, 2015, pp. 25–32. (cited on pages 29 and 33)

175. C. Zhu, G. Li, W. Wang, and R. Wang, "An innovative salient object detection using center-dark channel prior," in *Int. Conf. Comput. Vis. Worksh.*, 2017. (cited on pages 29 and 33)

176. D. Feng, N. Barnes, S. You, and C. McCarthy, "Local background enclosure for RGB-D salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 2343–2350. (cited on pages 29 and 33)

177. R. Cong, J. Lei, C. Zhang, Q. Huang, X. Cao, and C. Hou, "Saliency detection for stereoscopic images based on depth confidence analysis and multiple cues fusion," *IEEE Signal Process. Lett.*, vol. 23, no. 6, pp. 819–823, 2016. (cited on pages 29, 31, 33, and 58)

178. H. Song, Z. Liu, H. Du, G. Sun, O. Le Meur, and T. Ren, "Depth-aware salient object detection and segmentation via multiscale discriminative saliency fusion and bootstrap learning," *IEEE T. Image Process.*, vol. 26, no. 9, pp. 4204–4216, 2017. (cited on pages 29 and 33)

179. J. Guo, T. Ren, and J. Bei, "Salient object detection for rgb-d image via saliency evolution," in *Int. Conf. Multimedia and Expo*, 2016, pp. 1–6. (cited on pages 29, 31, and 33)

180. S. Chen, X. Tan, B. Wang, and X. Hu, "Reverse attention for salient object detection," in *Eur. Conf. Comput. Vis.*, 2018. (cited on page 31)

181. T. Wang, L. Zhang, S. Wang, H. Lu, G. Yang, X. Ruan, and A. Borji, "Detect globally, refine locally: A novel approach to saliency detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 3127–3135. (cited on pages 31, 37, 63, 70, 86, 100, 101, and 111)

182. B. Settles, "Active learning literature survey," University of Wisconsin-Madison Department of Computer Sciences, Tech. Rep., 2009. (cited on page 31)

183. Y. Wei, F. Wen, W. Zhu, and J. Sun, "Geodesic saliency using background priors," in *Eur. Conf. Comput. Vis.*, 2012, pp. 29–42. (cited on pages 31, 99, and 103)

184. F. Liang, L. Duan, W. Ma, Y. Qiao, Z. Cai, and L. Qing, "Stereoscopic saliency model using contrast and depth-guided-background prior," *Neurocomputing*, vol. 275, pp. 2227–2238, 2018. (cited on pages 33 and 55)

185. S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from RGB-D images for object detection and segmentation," in *Eur. Conf. Comput. Vis.*, 2014, pp. 345–360. (cited on pages 35 and 37)

186. D. Du, L. Wang, H. Wang, K. Zhao, and G. Wu, "Translate-to-recognize networks for rgb-d scene recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 11 836–11 845. (cited on page 37)

187. C. M. Bishop, "Training with noise is equivalent to tikhonov regularization," *Neural Computation*, vol. 7, no. 1, pp. 108–116, 1995. (cited on page 37)

188. J. Wei, S. Wang, Z. Wu, C. Su, Q. Huang, and Q. Tian, "Label decoupling framework for salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 13 025–13 034. (cited on pages 37, 61, 63, and 70)

189. Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 1155–1162. (cited on pages 38, 70, 86, 87, 99, 107, and 117)

190. M.-M. Cheng, N. J. Mitra, X. Huang, and S.-M. Hu, "Salientshape: group saliency in image collections," *The Vis. Comput.*, vol. 30, no. 4, pp. 443–453, 2014. (cited on pages 38, 70, 86, 99, and 117)

191. D.-P. Fan, M.-M. Cheng, J.-J. Liu, S.-H. Gao, Q. Hou, and A. Borji, "Salient objects in clutter: Bringing salient object detection to the foreground," in *Eur. Conf. Comput. Vis.*, 2018. (cited on pages 38, 52, 70, and 117)

192. A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *Adv. Neural Inform. Process. Syst.*, 2017. (cited on pages 39 and 128)

193. C. Liu, J. Yuen, and A. Torralba, "Sift flow: Dense correspondence across scenes and its applications," *IEEE T. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 978–994, 2010. (cited on page 42)

194. Z. Zhang, "Microsoft kinect sensor and its effect," *IEEE multimedia*, vol. 19, no. 2, pp. 4–10, 2012. (cited on page 42)

195. D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles," in *IEEE Conf. Comput. Vis. Pattern Recog.* IEEE, 2010, pp. 2432–2439. (cited on page 42)

196. R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan, "Light field photography with a hand-held plenoptic camera," Ph.D. dissertation, Stanford University, 2005. (cited on page 42)

197. M. W. Tao, S. Hadap, J. Malik, and R. Ramamoorthi, "Depth from combining defocus and correspondence using light-field cameras," in *Int. Conf. Comput. Vis.*, 2013, pp. 673–680. (cited on page 42)

198. Y. Hua, P. Kohli, P. Uplavikar, A. Ravi, S. Gunaseelan, J. Orozco, and E. Li, "Holopix50k: A large-scale in-the-wild stereo image dataset," in *CVPR Workshop on Computer Vision for Augmented and Virtual Reality, Seattle, WA, 2020.*, June 2020. (cited on pages 42, 43, 48, and 58)

199. M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE T. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 569–582, 2015. (cited on pages 43 and 48)

200. H. Huang, z. li, R. He, Z. Sun, and T. Tan, "Introvae: Introspective variational autoencoders for photographic image synthesis," in *Adv. Neural Inform. Process. Syst.*, vol. 31, 2018, pp. 52–63. (cited on page 45)

201. A. Pajot, E. de Bezenac, and P. Gallinari, "Unsupervised adversarial image reconstruction," in *Int. Conf. Learn. Represent.*, 2019. (cited on page 45)

202. A. Grover and S. Ermon, "Uncertainty autoencoders: Learning compressed representations via variational information maximization," in *Proceedings of Machine Learning Research*, vol. 89, 2019, pp. 2514–2524. (cited on page 45)

203. J. Wang, Y. Zhong, Y. Dai, K. Zhang, P. Ji, and H. Li, "Displacement-invariant matching cost learning for accurate optical flow estimation," in *Adv. Neural Inform. Process. Syst.*, 2020. (cited on pages 48 and 49)

204. J. Kim, D. Han, Y.-W. Tai, and J. Kim, "Salient region detection via high-dimensional color transform and local spatial support," *IEEE T. Image Process.*, vol. 25, no. 1, pp. 9–23, 2015. (cited on page 49)

205. B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE T. Pattern Anal. Mach. Intell.*, 2017. (cited on page 50)

206. X. Zhao, L. Zhang, Y. Pang, H. Lu, and L. Zhang, "A single stream network for robust and real-time rgb-d salient object detection," in *Eur. Conf. Comput. Vis.*, 2020. (cited on page 55)

207. H. Kim, S. Lee, and A. C. Bovik, "Saliency prediction on stereoscopic videos," *IEEE T. Image Process.*, vol. 23, no. 4, pp. 1476–1490, 2014. (cited on page 58)

208. Y. Fang, J. Wang, M. Narwaria, P. Le Callet, and W. Lin, "Saliency detection for stereoscopic images," in *2013 Visual Communications and Image Processing (VCIP)*, 2013, pp. 1–6. (cited on page 58)

209. Q. Zhang, X. Wang, S. Wang, S. Li, S. Kwong, and J. Jiang, "Learning to explore intrinsic saliency for stereoscopic video," in *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2019. (cited on page 58)

210. K. K. Singh and Y. J. Lee, "Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization," in *Int. Conf. Comput. Vis.*, 2017. (cited on page 60)

211. C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Int. Conf. Mach. Learn.*, 2017, pp. 1321–1330. (cited on page 61)

212. W. Wang, S. Zhao, J. Shen, S. C. Hoi, and A. Borji, "Salient object detection with pyramid attention and salient edges," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 1448–1457. (cited on pages 63, 92, and 94)

213. R. M. Neal, "MCMC using hamiltonian dynamics," *Handbook of Markov Chain Monte Carlo*, vol. 54, pp. 113–162, 2010. (cited on pages 64 and 65)

214. P. Knöbelreiter, C. Reinbacher, A. Shekhovtsov, and T. Pock, "End-to-end training of hybrid cnn-crf models for stereo," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 1456–1465. (cited on page 74)

215. J.-X. Zhao, J.-J. Liu, D.-P. Fan, Y. Cao, J. Yang, and M.-M. Cheng, "Egnet: Edge guidance network for salient object detection," in *Int. Conf. Comput. Vis.*, 2019, pp. 8779–8788. (cited on pages 78, 94, 107, 111, and 118)

216. J. Zhang, B. Li, Y. Dai, F. Porikli, and M. He, "Integrated deep and shallow networks for salient object detection," in *IEEE Int. Conf. Image Process.*, 2017, pp. 1537–1541. (cited on page 78)

217. J. Zhang, Y. Dai, and F. Porikli, "Deep salient object detection by integrating multi-level cues," in *IEEE Winter Conf. App. Comput. Vis.*, 2017, pp. 1–10. (cited on page 78)

218. P. Siva, C. Russell, T. Xiang, and L. Agapito, "Looking beyond the image: Unsupervised learning for object saliency and detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 3238–3245. (cited on page 78)

219. M. Tang, F. Perazzi, A. Djelouah, I. B. Ayed, C. Schroers, and Y. Boykov, "On regularized losses for weakly-supervised cnn segmentation," in *Eur. Conf. Comput. Vis.*, 2018, pp. 524–540. (cited on page 79)

220. D. Lin, J. Dai, J. Jia, K. He, and J. Sun, "Scribblesup: Scribble-supervised convolutional networks for semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 3159–3167. (cited on pages 79, 80, and 90)

221. B. Wang, G. Qi, S. Tang, T. Zhang, Y. Wei, L. Li, and Y. Zhang, "Boundary perception guidance: A scribble-supervised semantic segmentation approach," in *IJCAI*, 2019, pp. 3663–3669. (cited on pages 79, 80, and 90)

222. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 248–255. (cited on pages 79 and 85)

223. T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. Zitnick, "Microsoft coco: Common objects in context," in *Eur. Conf. Comput. Vis.*, 2014, pp. 740–755. (cited on pages 79 and 117)

224. S. Xie and Z. Tu, "Holistically-nested edge detection," in *Int. Conf. Comput. Vis.*, 2015, pp. 1395–1403. (cited on page 81)

225. Y. Liu, M.-M. Cheng, X. Hu, K. Wang, and X. Bai, "Richer convolutional features for edge detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 3000–3009. (cited on pages 82 and 89)

226. P. Zhang, D. Wang, H. Lu, H. Wang, and B. Yin, "Learning uncertain convolutional features for accurate saliency detection," in *Int. Conf. Comput. Vis.*, 2017, pp. 212–221. (cited on page 86)

227. Z. Deng, X. Hu, L. Zhu, X. Xu, J. Qin, G. Han, and P.-A. Heng, "R^3Net: recurrent residual refinement network for saliency detection," in *Int. Jt. Conf. Artif. Intell.*, 2018, pp. 684–690. (cited on page 86)

228. T. Zhao and X. Wu, "Pyramid feature attention network for saliency detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 3085–3094. (cited on page 86)

229. Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille, "The secrets of salient object segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 280–287. (cited on pages 86 and 117)

230. C. Rother, V. Kolmogorov, and A. Blake, "Grabcut -interactive foreground extraction using iterated graph cuts," *ACM Transactions on Graphics (SIGGRAPH)*, 2004. (cited on page 89)

231. B. Wang, Q. Chen, M. Zhou, Z. Zhang, X. Jin, and K. Gai, "Progressive feature polishing network for salient object detection," in *AAAI Conf. Art. Intell.*, 2020. (cited on pages 92 and 94)

232. W. Wang, J. Shen, X. Dong, and A. Borji, "Salient object detection driven by fixation prediction," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 1711–1720. (cited on page 92)

233. D. Li, C. Rodriguez, X. Yu, and H. Li, "Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison," in *IEEE Winter Conf. App. Comput. Vis.*, 2020. (cited on page 92)

234. J. Xie, R. Gao, Z. Zheng, S.-C. Zhu, and Y. N. Wu, "Learning dynamic generator model by alternating back-propagation through time," in *AAAI Conf. Art. Intell.*, vol. 33, 2019, pp. 5498–5507. (cited on pages 93 and 105)

235. Y. Zhu, J. Xie, B. Liu, and A. Elgammal, "Learning feature-to-feature translator by alternating back-propagation for generative zero-shot learning," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 9844–9854. (cited on page 93)

236. X. Li, F. Yang, H. Cheng, W. Liu, and D. Shen, "Contour knowledge transfer for salient object detection," in *Eur. Conf. Comput. Vis.*, 2018. (cited on pages 93, 94, 100, and 111)

237. P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2014. (cited on pages 93 and 94)

238. S. E. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich, "Training deep neural networks on noisy labels with bootstrapping," in *Int. Conf. Learn. Represent.*, 2014. (cited on page 95)

239. R. Tanno, A. Saeedi, S. Sankaranarayanan, D. C. Alexander, and N. Silberman, "Learning from noisy labels by regularized estimation of annotator confusion," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. (cited on page 95)

240. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. (cited on pages 97 and 100)

241. J. Xie, R. Gao, E. Nijkamp, S.-C. Zhu, and Y. N. Wu, "Representation learning: A statistical perspective," *Annual Review of Statistics and Its Application*, vol. 7, pp. 303–335, 2020. (cited on page 98)

242. J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. (cited on page 99)

243. Baidu, "PaddlePaddle," `https://www.paddlepaddle.org.cn`. (cited on page 100)

244. C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *Int. Conf. Learn. Represent.*, 2017. (cited on pages 101, 107, 112, and 119)

245. A. Borji, "Saliency Prediction in the Deep Learning Era: An Empirical Investigation," *ArXiv e-prints*, Oct. 2018. (cited on pages 107 and 111)

246. L. Zhang, J. Zhang, Z. Lin, H. Lu, and Y. He, "Capsal: Leveraging captioning to boost semantics for salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. (cited on page 107)

247. N. Tong, H. Lu, X. Ruan, and M. H. Yang, "Salient object detection via bootstrap learning," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 1884–1892. (cited on pages 107 and 108)

248. T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang, "Learning from massive noisy labeled data for image classification," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 2691–2699. (cited on page 108)

249. S. Guo, W. Huang, H. Zhang, C. Zhuang, D. Dong, M. R. Scott, and D. Huang, "Curriculumnet: Weakly supervised learning from large-scale web images," in *Eur. Conf. Comput. Vis.*, 2018. (cited on pages 108 and 113)

250. K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Int. Conf. Comput. Vis.*, 2017. (cited on pages 108 and 127)

251. A. Roy and S. Todorovic, "Monocular depth estimation using neural regression forest," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 5506–5514. (cited on page 108)

252. M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981. (cited on pages 108, 109, and 113)

253. Y. Siddiqui, J. Valentin, and M. Niessner, "Viewal: Active learning with viewpoint entropy for semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. (cited on page 109)

254. D. Acuna, A. Kar, and S. Fidler, "Devil is in the edges: Learning semantic boundaries from noisy annotations," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. (cited on pages 109, 121, and 127)

255. Y. Pang, X. Zhao, L. Zhang, and H. Lu, "Multi-scale interactive network for salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. (cited on pages 111 and 118)

256. D.-P. Fan, W. Wang, M.-M. Cheng, and J. Shen, "Shifting more attention to video salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2019. (cited on page 111)

257. A. I. Schein and L. H. Ungar, "Active learning for logistic regression: an evaluation," *Machine Learning*, vol. 68, no. 3, pp. 235–265, 2007. (cited on page 113)

258. D. Meng and Q. Zhao, "What objective does self-paced learning indeed optimize?" *CoRR*, vol. abs/1511.06049, 2015. (cited on page 114)

259. T. Pi, X. Li, Z. Zhang, D. Meng, F. Wu, J. Xiao, and Y. Zhuang, "Self-paced boost learning for classification," in *Int. Jt. Conf. Artif. Intell.*, 2016, pp. 1932–1938. (cited on page 114)

260. M. P. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," in *Adv. Neural Inform. Process. Syst.*, 2010, pp. 1189–1197. (cited on pages 114 and 115)

261. A. Katharopoulos and F. Fleuret, "Not all samples are created equal: Deep learning with importance sampling," in *Int. Conf. Mach. Learn.*, 2018, pp. 2525–2534. (cited on page 115)

262. W. Hu, Y. Yang, W. Zhang, and Y. Xie, "Moving object detection using tensor-based low-rank and saliently fused-sparse decomposition," *IEEE T. Image Process.*, vol. 26, no. 2, pp. 724–737, 2017. (cited on page 116)

263. A. Bordes, S. Ertekin, J. Weston, and L. Bottou, "Fast kernel classifiers with online and active learning," *J. Mach. Learn. Research*, vol. 6, pp. 1579–1619, 2005. (cited on page 120)

264. M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient langevin dynamics," in *Int. Conf. Mach. Learn.*, 2011, pp. 681–688. (cited on page 126)

265. R. Garg, B. V. Kumar, G. Carneiro, and I. Reid, "Unsupervised cnn for single view depth estimation: Geometry to the rescue," in *Eur. Conf. Comput. Vis.* Springer, 2016, pp. 740–756. (cited on page 127)

266. C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE T. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, 2016. (cited on page 127)

267. J. Jia, "Single image motion deblurring using transparency," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2007, pp. 1–8. (cited on page 127)