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

Jing Zhang[1,3,4⋆], Jianwen Xie[2], and Nick Barnes[1]

[1] Australian National University, Australia
[2] Cognitive Computing Lab, Baidu Research, USA
[3] Australian Centre for Robotic Vision, Australia
[4] Data61, Australia

Abstract. In this paper, 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. 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.

Keywords: Noisy saliency, Latent variable model, Langevin dynamics, Alternating back-propagation

1 Introduction

Visual saliency detection aims to locate salient regions that attract human attention. Conventional saliency detection methods [59,46] 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 [19,54,41,55,40,7,25,30,38,21,35,34,33,51], outperforming handcrafted feature-based solutions by a wide margin. However, the success

[⋆] Work was done while Jing Zhang was an intern mentored by Jianwen Xie.

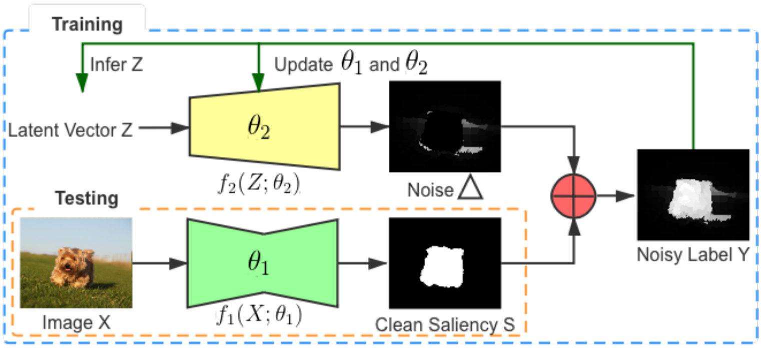

Fig. 1. An illustration of our framework. Representation: Each noisy label Y is represented as a sum of a clean saliency S and a noise map Δ. The clean saliency S is predicted from an image X by an encoder-decoder network f_1, and the noise is produced from a Gaussian noise vector Z by a generator network 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.

of deep models mainly depends on a large amount of accurate human labeling [31,3,15], which is typically expensive and time-consuming.

To relieve the burden of pixel-wise labeling, weakly supervised [17,31,52] and unsupervised saliency detection models [53,50,24] 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. In this paper, 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_i and Y_i are image and its corresponding noisy saliency label, we intend to disentangle noise Δ_i and clean saliency S_i from each noisy label Y_i, and learn a clean saliency predictor $f_1 : X_i \rightarrow S_i$. To achieve this, we propose a conditional latent variable model, which is a disentangled representation of noisy saliency Y_i. See Figure 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 $\varDelta_i$ to its clean saliency map S_i that is dependent on its image X_i. Specifically, the model consists of two sub-models: (1) saliency predictor f_1: an encoder-decoder network that maps an input image X_i to a latent clean saliency map S_i, and (2) noise generator f_2: a top-down neural network that produces a noise or error $\varDelta_i$ from a low-dimensional Gaussian latent vector Z_i.

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) [13] or alternating back-propagation (ABP) [9,44,60]. 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). In this paper, 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 $\varDelta$ such that $S + \varDelta$ perfectly matches the provided noisy label Y, we further adopt the edge-aware smoothness loss [37] to serve as a regularization to force each latent saliency map S_i to have a similar structure as its input image X_i. 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 [53,50,24,18] in the following aspects: Firstly, unlike [53], 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 $\varDelta_i$ as a non-linear transformation of low-dimensional Gaussian noise Z_i, which is a latent variable that need to be inferred during training, while [53,50,24,18] have no noise inference process. Thirdly, we have no constraints on the number of noisy labels generated from each image, while [53,50,24] 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 [18], where object edges are used to produce pseudo saliency labels via multi-scale combinatorial grouping (MCG) [1].

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 latent variable 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.

2 Related Work

Fully supervised saliency detection models [30,38,21,35,34,25,41,36,58,57] mainly focus on designing networks that utilize image context information, multi-scale information, and image structure preservation. [30] introduces feature polishing modules to update each level of features by incorporating all higher levels of context information. [38] 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. [34] designs a pyramid attention structure with an edge detection module to perform edge-preserving salient object detection. [25] uses a hybrid loss for boundary-aware saliency detection. [36] proposes to use the stacked pyramid attention, which exploits multi-scale saliency information, along with an edge-related loss for 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 [31,17,48], noisy labels [53,50,24], object contours [18], scribble annotations [52], *etc.*[31] 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. [50] fuses saliency maps from unsupervised handcrafted feature-based methods with heuristics within a deep learning framework. [53] collaboratively updates a saliency prediction module and a noise module to achieve learning saliency from multiple noisy labels. In [24], the initial noisy labels are refined by a self-supervised learning technique, and then treated as pseudo labels. [18] creates a contour-to-saliency network, where saliency masks are generated by its contour detection branch via MCG [1] 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 [26,47]; (2) estimating the noise distribution by assuming that noisy labels are corrupted from clean labels by an unknown noise transition matrix [8,29] and (3) training on selected samples [12,20]. [26] deals with noisy labeling by augmenting the prediction objective with a notion of perceptual consistency. [47] proposes a framework to solve noisy label problem by updating both model parameters and labels. [29] 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. [12] 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.

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 top-down generator network that produces a noise or error $\mathit{\Delta}$ 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 the image X and outputs clean saliency S, is our desired model for saliency detection.

3.1 Noise-Aware Encoder-Decoder Network

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

$$S = f_1(X; \theta_1), \tag{1}$$

$$\mathit{\Delta} = f_2(Z; \theta_2), Z \sim \mathcal{N}(0, I_d), \tag{2}$$

$$Y = S + \mathit{\Delta} + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2 I_D), \tag{3}$$

where f_1 in Eq. (1) is an encoder-decoder structure parameterized by θ_1 for saliency detection. It takes image X as input and predicts the clean saliency map S. Eq. (2) defines a noise generator, where Z is a 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 the noise $\mathit{\Delta}$ from Z. In Eq. (3), we assume that the observed noisy label Y is a sum of the clean saliency map S and the noise $\mathit{\Delta}$, 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 Gaussian noise, the generated noise $\mathit{\Delta}$ is not necessary Gaussian due to the non-linear transformation f_2

We call our network the noise-aware encoder-decoder network as it explicitly decomposes the noisy labels Y into noise $\mathit{\Delta}$ and clean labels S, and simultaneously learns a mapping from image X to clean saliency S via an encoder-decoder network as shown in Fig. 1. Since the resulting model involves latent variables Z, training the model by maximum likelihood learning 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.

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 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 labels 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:

$$\begin{aligned} \frac{\partial}{\partial\theta} \log p_\theta(Y|X) &= \frac{1}{p_\theta(Y|X)} \frac{\partial}{\partial\theta} p_\theta(Y|X) \\ &= \mathrm{E}_{p_\theta(Z|Y,X)} \left[\frac{\partial}{\partial\theta} \log p_\theta(Y, Z|X) \right]. \end{aligned} \tag{4}$$

The expectation term $\mathrm{E}_{p_\theta(Z|Y,X)}$ is analytically intractable. The conventional way of training such a latent variable model is to approximate the above expectation term with another family of posterior distribution $p_\phi(Z|Y, X)$, such as variational inference. In this paper, 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 [23] (a gradient-based Monte Carlo method) to sample Z. The Langevin Dynamics for sampling $Z \sim p_\theta(Z|Y, X)$ iterates:

$$Z_{t+1} = Z_t + \frac{s^2}{2} \left[\frac{\partial}{\partial Z} \log p_\theta(Y, Z_t|X) \right] + s\mathcal{N}(0, I_d), \tag{5}$$

with

$$\frac{\partial}{\partial Z} \log p_\theta(Y, Z|X) = \frac{1}{\sigma^2}(Y - f(X, Z; \theta)) \frac{\partial}{\partial Z} f(X, Z) - Z, \tag{6}$$

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:

$$\begin{aligned} \frac{\partial}{\partial\theta} \mathcal{L}(\theta) &\approx \sum_{i=1}^n \frac{\partial}{\partial\theta} \log p_\theta(Y_i, X_i|Z_i), \\ &= \sum_{i=1}^n \frac{1}{\sigma^2}(Y_i - f(X_i, Z_i; \theta)) \frac{\partial}{\partial\theta} f(X_i, Z_i). \end{aligned} \tag{7}$$

Algorithm 1 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 $\gamma$
Output: Network parameters $\theta = \{\theta_1, \theta_2\}$, and the inferred latent vectors $\{Z_i\}_{i=1}^n$
Initialize $\theta_1$ with VGG16-Net[27] for image classification, $\theta_2$ with a truncated Gaussian distribution, and $Z_i$ with a standard Gaussian distribution.
for $k = 1, ..., K$ do
    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. (5), with $Z_i$ initialized as Gaussian white noise or the result from previous iteration.
    Learning back-propagation: Update model parameters $\theta$ by Adam [14] optimizer with a learning rate $\gamma$ 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. (7).
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 [37] 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 [37], we use first-order derivatives (*i.e.*, edge information) of both the latent clean saliency map S and image X to compute the smoothness loss

$$l_s(X, S) = \sum_{u,v} \sum_{d \in x,y} \Psi(|\partial_d S_{u,v}| e^{-\alpha|\partial_d X_{u,v}|}), \tag{8}$$

where Ψ is the Charbonnier penalty formula, defined as $\Psi(s) = \sqrt{s^2 + 1e^{-6}}$, (u, v) represents pixel coordinate, 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. (8).

The whole process of updating both $\{Z_i\}$ and $\theta = \{\theta_1, \theta_2\}$ is summarized in Algorithm 1, which is implemented as alternating back-propagation, because both gradients in Eq. (5) and (7) can be computed via back-propagation.

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. 4 is approximated by a tractable $q_\phi(Z|Y, X)$, such as $q_\phi(Z|Y, X) \sim \mathcal{N}(\mu_\phi(Y, X), \mathrm{diag}(v_\phi(Y, X)))$, 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:

$$\begin{aligned} &\min_\theta \min_\phi \mathrm{KL}(q_{\mathrm{data}}(Y|X) p_\phi(Z|Y, X) \| p_\theta(Z, Y|X)) = \\ &\min_\theta \min_\phi \mathrm{KL}(q_{\mathrm{data}}(Y|X) \| p_\theta(Y|X)) + \mathrm{KL}(p_\phi(Z|Y, X) \| p_\theta(Z|Y, X)). \end{aligned} \tag{9}$$

Recall that the maximum likelihood learning in our algorithm is equivalent to minimizing $\mathrm{KL}(q_{\mathrm{data}}(Y|X) \| p_\theta(Y|X))$, where $q_{\mathrm{data}}(Y|X)$ is the conditional training data distribution. The accuracy of variational inference in Eq. 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 $\mathrm{KL}(p_\phi(Z|Y,X)\|p_\theta(Z|Y,X))=0$. 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 [43] for a comprehensive tutorial on latent variable models.

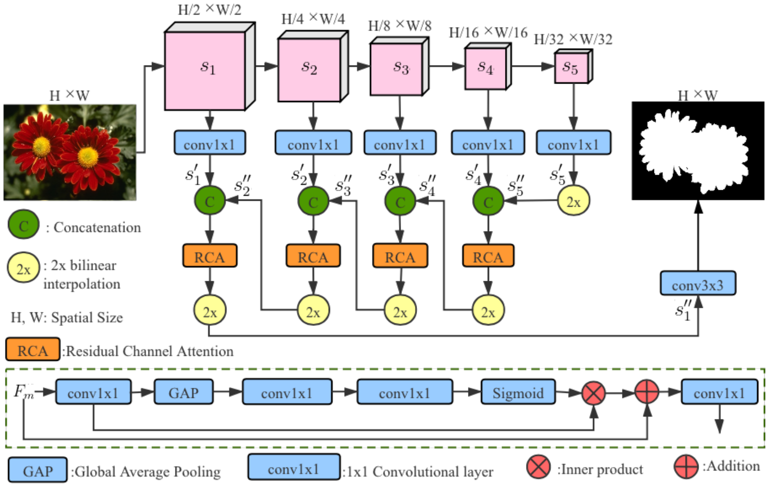

Fig. 2. An illustration of the encoder-decoder-based saliency detection network (Green part in Fig.1).

3.4 Network Architectural Design

We now introduce the architectural designs of the encoder-decoder network (f_1 in Eq. 1, or the green encoder-decoder in Fig. 1) and the noise generator network (f_2 in Eq. 2, or the yellow decoder in Fig. 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 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 VGG16-Net [27], ResNet [10], etc. Due to stride operations and multiple pooling layers 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 VGG16-Net [27] as backbone shown in Fig. 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, 1×1 convolutional layer is used to transform each s_m to a feature map s'_m of channel dimension 32. Then a Residual Channel Attention (RCA) module [56] is adopted to effectively fuse intermediate high- and low-level features as shown in Fig.2. Specifically, given 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 size 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 [56], we feed F_m to the channel attention block to achieve discriminative feature extraction. Inside each channel attention block, we perform "squeeze and excitation" [11] by first "squeezing" the feature map F_m to have half channel size to obtain better nonlinear interactions across channels, and then "exciting" the squeezed feature map to have the original channel size, as shown in Fig. 2. By adding one 3×3 convolutional layer to the lowest level RCA module, we obtain a one-channel saliency map $S_i = f_1(X_i; \theta_1)$.

4 Experiments

4.1 Experimental Setup

Datasets: We evaluate our performance on five saliency benchmark datasets. We used 10,553 images from the DUTS dataset [31] for training, and generate noisy labels from images using handcrafted feature based-methods, such as RBD [59], MR [46] and GS [39] due to their high efficiencies. Testing datasets include DUTS testing set, ECSSD [45], DUT [46], HKU-IS [16] and THUR [4].

Evaluation Metrics: Four evaluation metrics are used to evaluate the performance of ours and competing methods, including two widely used metrics: Mean Absolute Error ($\mathcal{M}$), mean F-measure (F_β), and two newly released structure-aware metrics: mean E-measure (E_ξ) [6] and S-measure (S_α) [5].

Training Details: Each input image is rescaled to 352×352 pixels. The encoder part in Fig. 2 is initialized using the VGG16 weights pretrained for image classification [27]. The weights of other layers are initialized using the "truncated Gaussian" policy, and the biases are initialized to be zeros. We use the Adam [14] optimizer with a momentum equal to 0.9, and decrease the learning rate by 10% after running 80% of the maximum epochs, which is 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.(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 [2] deep learning platform.

Table 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 [32]	NLDF [22]	MSNet [40]	CPD [41]	AFNet [7]	SCRN [42]	BASNet [25]	C2S [18]	WSI [17]	WSS [31]	MNL [53]	MSW [48]	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**

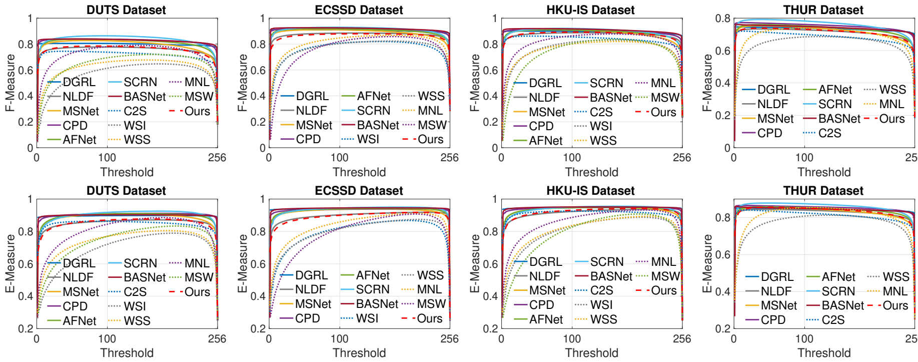

Fig. 3. F-measure and E-measure curves on four datasets (DUTS, ECSSD, HKU-IS, THUR). Best viewed on screen.

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 1 and Fig. 3. Table 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

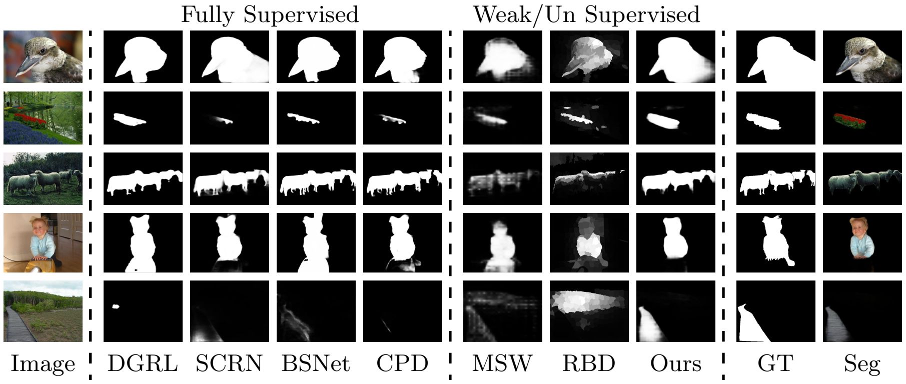

Fig. 4. Comparison of saliency predictions, where each row displays an input image, its predicted saliency maps by four fully supervised competing methods (DGRL, SCRN, BASNet, and CPD), one weakly (MSW) and one unsupervised (RBD) methods, our prediction (Ours), the ground truth (GT) saliency map and our segmented foreground image (Seg).

method achieves an approximately 2% performance improvement for S-measure, and 4% improvement for mean F-measure. Further, the proposed method even achieves comparable performance with some newly released fully supervised models. For example, we achieve comparable performance with NLDF [22] and DGRL [32] on all the five benchmark datasets. Fig.3 shows the 256-dimensional F-measure and E-measure (where the x-axis represents threshold for saliency map binarization) of our method and competing methods on four datasets, where the weakly supervised/unsupervised methods are represented by dotted curves. We can observe that performance of fully supervised models is better than the weakly supervised/unsupervised models. As shown in Fig.3, our performance shows stability with different thresholds relative to existing methods, indicating the robustness of our proposed solution.

Figure 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. 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.

4.3 Ablation Study

We conduct the following experiments for 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. 2) directly learned from the noisy labels, without noise modeling. The performance is shown

Table 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 $\mathcal{M}$. ↑ & ↓ denote larger and smaller is better, respectively.

	DUTS				ECSSD				DUT				HKU-IS				THUR			
	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	$\mathcal{M}$
Model	↑	↑	↑	↓	↑	↑	↑	↓	↑	↑	↑	↓	↑	↑	↑	↓	↑	↑	↑	↓
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

in Table 2 with a label "f_1", which is clearly worse than our results. This result is also consistent with the conclusion that deep neural networks is not robust to noise [49].

(2) Encoder-decoder f_1 + smoothness loss l_s: As an extension of method "f_1", one can add the smoothness loss in Eq. (8) as a regularization to better use image prior information. We show its performance with a label "f_1 & l_s" in Table 2. We observe 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 2 as "f & l_c", which is better than or comparable with the existing weakly supervised/unsupervised methods shown in Table 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. 8 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 saliency boundaries usually are aligned with image boundaries, but they are not exactly the same. A soft and indirect penalty for edge dissimilarity seems to be more useful.

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 [25]. We treat the outputs as noisy labels, on which we train our

Table 3. Experimental results for model analysis. ↑ & ↓ denote larger and smaller is better, respectively.

	DUTS				ECSSD				DUT				HKU-IS				THUR			
	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	$\mathcal{M}$
Model	↑	↑	↑	↓	↑	↑	↑	↓	↑	↑	↑	↓	↑	↑	↑	↓	↑	↑	↑	↓
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

model. The performances are shown in Table 3 as f-BAS. By comparing the performances of f-BAS with those of BASNet in Table 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 [59], MR [46] and GS [39], respectively. As our method has no constraints on the number of noisy labels per image, we conduct experiments to test our models learned in the setting of one noisy label per image. In Table 3, we report the performances of the models learned from noisy labels generated by RBD [59], MR [46] and GS [39], respectively. We use f-RBD, f-MR and f-GS to represent the corresponding results. We observe comparable performances with those using the setting of multiple noisy labels per image, which indicates our method is robust to the number of noisy labels per image and different levels of noisy labels. (RBD ranks the 1^{st} among unsupervised saliency detection models in [3]. RBD, MR and GS can represent different levels of noisy labels). We also show in Table 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 labels can be treated as special cases of noisy labels 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 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 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: In this paper, 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.(4). One can also train the model in a conditional variational inference framework [28] as shown in Eq. (9). Following cVAE [28], 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.(8). We present the cVAE results in Table 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.

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 [50,53,24], 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 predictive model is an encoder-decoder framework, while the noise is modeled as a non-linear transformation of Gaussian latent variables, 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 alternating back-propagation algorithm [9,44], which is superior to variational inference. Extensive experiments conducted on different benchmark datasets demonstrate the state-of-the-art performances of our model among the unsupervised saliency detection methods.

Acknowledgments. This research was supported in part by the Australia Research Council Centre of Excellence for Robotics Vision (CE140100016).

References

1. Arbeláez, P., Pont-Tuset, J., Barron, J., Marques, F., Malik, J.: Multiscale combinatorial grouping. In: IEEE Conference on Computer Vision and Pattern Recognition (2014) 3, 4
2. Baidu: PaddlePaddle. `https://www.paddlepaddle.org.cn` 9
3. Borji, A., Cheng, M.M., Jiang, H., Li, J.: Salient object detection: A benchmark. IEEE Transactions on Image Processing **24**(12), 5706–5722 (2015) 2, 13
4. Cheng, M.M., Mitra, N., Huang, X., Hu, S.M.: Salientshape: group saliency in image collections. The Visual Computer **30**(4), 443–453 (2014) 9
5. Fan, D.P., Cheng, M.M., Liu, Y., Li, T., Borji, A.: Structure-measure: A new way to evaluate foreground maps. In: International Conference on Computer Vision. pp. 4548–4557 (2017) 9
6. Fan, D.P., Gong, C., Cao, Y., Ren, B., Cheng, M.M., Borji, A.: Enhanced-alignment Measure for Binary Foreground Map Evaluation. In: International Joint Conference on Artificial Intelligence. pp. 698–704 (2018) 9
7. Feng, M., Lu, H., Ding, E.: Attentive feedback network for boundary-aware salient object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (2019) 1, 10
8. Goldberger, J., Ben-Reuven, E.: Training deep neural networks using a noise adaptation layer. In: International Conference on Learning Representations (2017) 4
9. Han, T., Lu, Y., Zhu, S.C., Wu, Y.N.: Alternating back-propagation for generator network. In: AAAI Conference on Artificial Intelligence (2017) 3, 14
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016) 8
11. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: IEEE Conference on Computer Vision and Pattern Recognition (2018) 9
12. Jiang, L., Zhou, Z., Leung, T., Li, L.J., Fei-Fei, L.: Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In: International Conference on Machine Learning (2018) 4, 5
13. Kingma, D., Welling, M.: Auto-encoding variational bayes. In: International Conference on Learning Representations (2014) 3
14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 7, 9
15. Li, D., Rodriguez, C., Yu, X., Li, H.: Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In: IEEE Winter Conference on Applications of Computer Vision (2020) 2
16. Li, G., Yu, Y.: Visual saliency based on multiscale deep features. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 5455–5463 (2015) 9
17. Li, G., Xie, Y., Lin, L.: Weakly supervised salient object detection using image labels. In: AAAI Conference on Artificial Intelligence (2018) 2, 4, 10
18. Li, X., Yang, F., Cheng, H., Liu, W., Shen, D.: Contour knowledge transfer for salient object detection. In: European Conference on Computer Vision (2018) 3, 4, 10
19. Liu, N., Han, J., Yang, M.H.: Picanet: Learning pixel-wise contextual attention for saliency detection. In: IEEE Conference on Computer Vision and Pattern Recognition (2018) 1
20. Liu, T., Tao, D.: Classification with noisy labels by importance reweighting. IEEE Transactions on Pattern Analysis and Machine Intelligence **38**(3), 447–461 (2016) 4

21. Liu, Y., Zhang, Q., Zhang, D., Han, J.: Employing deep part-object relationships for salient object detection. In: International Conference on Computer Vision (2019) 1, 4
22. Luo, Z., Mishra, A., Achkar, A., Eichel, J., Li, S., Jodoin, P.M.: Non-local deep features for salient object detection. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 6609–6617 (2017) 10, 11
23. Neal, R.M.: MCMC using hamiltonian dynamics. Handbook of Markov Chain Monte Carlo **54**, 113–162 (2010) 6
24. Nguyen, D.T., Dax, M., Mummadi, C.K., Ngo, T.P.N., Nguyen, T.H.P., Lou, Z., Brox, T.: Deepusps: Deep robust unsupervised saliency prediction with self-supervision. In: Advances in Neural Information Processing Systems (2019) 2, 3, 4, 14
25. Qin, X., Zhang, Z., Huang, C., Gao, C., Dehghan, M., Jagersand, M.: Basnet: Boundary-aware salient object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (2019) 1, 4, 10, 12
26. Reed, S.E., Lee, H., Anguelov, D., Szegedy, C., Erhan, D., Rabinovich, A.: Training deep neural networks on noisy labels with bootstrapping. In: International Conference on Learning Representations (2014) 4
27. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2014) 7, 8, 9
28. Sohn, K., Lee, H., Yan, X.: Learning structured output representation using deep conditional generative models. In: Advances in Neural Information Processing Systems. pp. 3483–3491 (2015) 14
29. Tanno, R., Saeedi, A., Sankaranarayanan, S., Alexander, D.C., Silberman, N.: Learning from noisy labels by regularized estimation of annotator confusion. In: IEEE Conference on Computer Vision and Pattern Recognition (2019) 4
30. Wang, B., Chen, Q., Zhou, M., Zhang, Z., Jin, X., Gai, K.: Progressive feature polishing network for salient object detection. In: AAAI Conference on Artificial Intelligence. pp. 12128–12135 (2020) 1, 4
31. Wang, L., Lu, H., Wang, Y., Feng, M., Wang, D., Yin, B., Ruan, X.: Learning to detect salient objects with image-level supervision. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 136–145 (2017) 2, 4, 9, 10
32. Wang, T., Zhang, L., Wang, S., Lu, H., Yang, G., Ruan, X., Borji, A.: Detect globally, refine locally: A novel approach to saliency detection. In: IEEE Conference on Computer Vision and Pattern Recognition (2018) 10, 11
33. Wang, W., Shen, J., Dong, X., Borji, A.: Salient object detection driven by fixation prediction. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1711–1720 (2018) 1
34. Wang, W., Zhao, S., Shen, J., Hoi, S.C.H., Borji, A.: Salient object detection with pyramid attention and salient edges. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1448–1457 (2019) 1, 4
35. Wang, W., Shen, J., Cheng, M.M., Shao, L.: An iterative and cooperative top-down and bottom-up inference network for salient object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (2019) 1, 4
36. Wang, W., Zhao, S., Shen, J., Hoi, S.C.H., Borji, A.: Salient object detection with pyramid attention and salient edges. In: IEEE Conference on Computer Vision and Pattern Recognition (2019) 4
37. Wang, Y., Yang, Y., Yang, Z., Zhao, L., Wang, P., Xu, W.: Occlusion aware unsupervised learning of optical flow. In: IEEE Conference on Computer Vision and Pattern Recognition (2018) 3, 7

38. Wei, J., Wang, S., Huang, Q.: F3net: Fusion, feedback and focus for salient object detection. In: AAAI Conference on Artificial Intelligence (2020) 1, 4
39. Wei, Y., Wen, F., Zhu, W., Sun, J.: Geodesic saliency using background priors. In: European Conference on Computer Vision. pp. 29–42 (2012) 9, 13
40. Wu, R., Feng, M., Guan, W., Wang, D., Lu, H., Ding, E.: A mutual learning method for salient object detection with intertwined multi-supervision. In: IEEE Conference on Computer Vision and Pattern Recognition (2019) 1, 10
41. Wu, Z., Su, L., Huang, Q.: Cascaded partial decoder for fast and accurate salient object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (2019) 1, 4, 10
42. Wu, Z., Su, L., Huang, Q.: Stacked cross refinement network for edge-aware salient object detection. In: International Conference on Computer Vision (2019) 10
43. Xie, J., Gao, R., Nijkamp, E., Zhu, S.C., Wu, Y.N.: Representation learning: A statistical perspective. Annual Review of Statistics and Its Application **7**, 303–335 (2020) 8
44. Xie, J., Gao, R., Zheng, Z., Zhu, S.C., Wu, Y.N.: Learning dynamic generator model by alternating back-propagation through time. In: AAAI Conference on Artificial Intelligence. vol. 33, pp. 5498–5507 (2019) 3, 14
45. Yan, Q., Xu, L., Shi, J., Jia, J.: Hierarchical saliency detection. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1155–1162 (2013) 9
46. Yang, C., Zhang, L., Lu, H., Ruan, X., Yang, M.: Saliency detection via graph-based manifold ranking. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 3166–3173 (2013) 1, 9, 13
47. Yi, K., Wu, J.: Probabilistic end-to-end noise correction for learning with noisy labels. In: IEEE Conference on Computer Vision and Pattern Recognition (2019) 4
48. Zeng, Y., Zhuge, Y., Lu, H., Zhang, L., Qian, M., Yu, Y.: Multi-source weak supervision for saliency detection. In: IEEE Conference on Computer Vision and Pattern Recognition (2019) 4, 10
49. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning requires rethinking generalization. In: International Conference on Learning Representations (2017) 12
50. Zhang, D., Han, J., Zhang, Y.: Supervision by fusion: Towards unsupervised learning of deep salient object detector. In: International Conference on Computer Vision (2017) 2, 3, 4, 14
51. Zhang, J., Fan, D.P., Dai, Y., Anwar, S., Saleh, F.S., Zhang, T., Barnes, N.: Uc-net: Uncertainty inspired rgb-d saliency detection via conditional variational autoencoders. In: IEEE Conference on Computer Vision and Pattern Recognition (2020) 1
52. Zhang, J., Yu, X., Li, A., Song, P., Liu, B., Dai, Y.: Weakly-supervised salient object detection via scribble annotations. In: IEEE Conference on Computer Vision and Pattern Recognition (2020) 2, 4
53. Zhang, J., Zhang, T., Dai, Y., Harandi, M., Hartley, R.: Deep unsupervised saliency detection: A multiple noisy labeling perspective. In: IEEE Conference on Computer Vision and Pattern Recognition (2018) 2, 3, 4, 10, 14
54. Zhang, P., Wang, D., Lu, H., Wang, H., Ruan, X.: Amulet: Aggregating multi-level convolutional features for salient object detection. In: International Conference on Computer Vision (2017) 1
55. Zhang, X., Wang, T., Qi, J., Lu, H., Wang, G.: Progressive attention guided recurrent network for salient object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (2018) 1

56. Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., Fu, Y.: Image super-resolution using very deep residual channel attention networks. In: European Conference on Computer Vision (2018) 9
57. Zhao, J.X., Cao, Y., Fan, D.P., Cheng, M.M., Li, X.Y., Zhang, L.: Contrast prior and fluid pyramid integration for rgbd salient object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (2019) 4
58. Zhao, J.X., Liu, J.J., Fan, D.P., Cao, Y., Yang, J., Cheng, M.M.: Egnet:edge guidance network for salient object detection. In: International Conference on Computer Vision (2019) 4
59. Zhu, W., Liang, S., Wei, Y., Sun, J.: Saliency optimization from robust background detection. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 2814–2821 (2014) 1, 9, 13
60. Zhu, Y., Xie, J., Liu, B., Elgammal, A.: Learning feature-to-feature translator by alternating back-propagation for generative zero-shot learning. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 9844–9854 (2019) 3