Uncertainty Inspired RGB-D Saliency Detection

Jing Zhang, Deng-Ping Fan, Yuchao Dai,
Saeed Anwar, Fatemeh Saleh, Sadegh Aliakbarian, and Nick Barnes

Abstract—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.

Index Terms—Uncertainty, RGB-D Saliency Prediction, Conditional Variational Autoencoders, Alternating Back-Propagation.

✦

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]–[9]. 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], [10]–[15]. 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.

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 [16]. Building upon large scale RGB-D datasets, deep convolutional neural network-based RGB-D saliency detection models [10], [11], [14], [17], [18] have made profound progress. We argue that the way RGB-D saliency detection progresses

- *Jing Zhang is with Research School of Engineering, the Australian National University, ACRV, DATA61-CSIRO. (Email: zjnwpu@gmail.com)*
- *Deng-Ping Fan is with the CS, Nankai University, China. (Email: dengpfan@gmail.com)*
- *Yuchao Dai is with School of Electronics and Information, Northwestern Polytechnical University, China. (Email: daiyuchao@gmail.com)*
- *Saeed Anwar is with the Australian National University, DATA61-CSIRO. (Email: saeed.anwar@data61.csiro.au)*
- *Fatemeh Saleh is with the Australian National University, ACRV. (Email: fatemehsadat.saleh@anu.edu.au)*
- *Sadegh Aliakbarian is with the Australian National University, ACRV. (Email: sadegh.aliakbarian@anu.edu.au)*
- *Nick Barnes is with Research School of Engineering, the Australian National University. (Email: nick.barnes@anu.edu.au)*
- *A preliminary version of this work appeared at CVPR 2020 [1].*
- *Corresponding author: Deng-Ping Fan.*

through the conventional pipelines [10], [11], [14], [17], [18] fails to capture the uncertainty in labeling the GT saliency maps.

According to research in human visual perception [19], visual saliency detection is subjective to some extent. Each person could have specific preferences [20] in labeling the saliency map (which has been discussed in user-specific saliency detection [21]). More precisely speaking, the GT labeling process is never a deterministic process, which is different from category-aware tasks, such as semantic segmentation [22], 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. 1.

In Fig. 1, we present the GT saliency map and other candidate salient regions (produced by our CVAE-based method, which will be introduced in Section 3.2) that may attract human attention. Fig. 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. 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 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.

---

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.
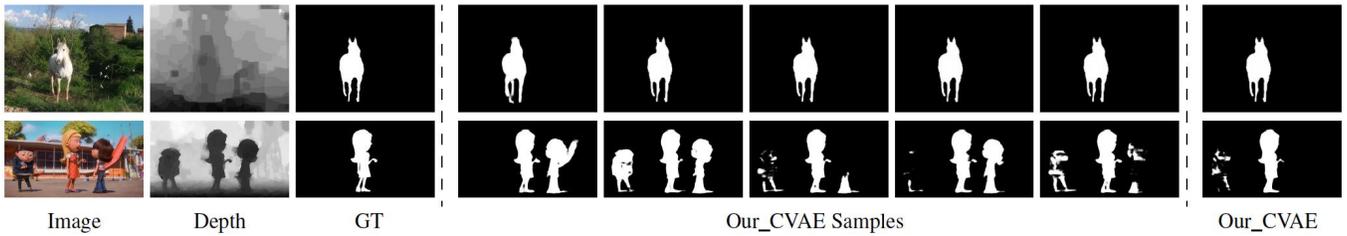
Fig. 1. GT compared with our predicted saliency maps. For simple context image (first row), we can produce consistent predictions. When dealing with complex scenarios where there exists uncertainties in salient regions (second row), our model can produce diverse predictions ("Our_CVAE Samples"), where "Our_CVAE" is our deterministic prediction after the saliency consensus module, which will be introduced in Section 3.3.

To infer the latent variable, we introduce two different strategies:

- A Conditional Variational Auto-encoder (CVAE) [23] based model with an additional encoder to approximate the posterior distribution of the latent variable.
- The Alternating Back-Propagation (ABP) [24] based technique, which directly samples the latent variable from the true posterior distribution via Langevin Dynamics based Markov chain Monte Carlo (MCMC) sampling [25], [26].

This paper is an extended version of our conference paper, UC-Net [1]. In particular, UC-Net focuses on generating saliency maps via CVAE and augmented ground-truth to model diversity and to avoid posterior collapse problem [27]. While UC-Net showed promising performance by modeling such variations, it still has a number of shortcomings. Firstly, UC-Net requires engineering efforts (ground-truth augmentation) to model diversity and achieve stabilized training (mitigating posterior collapse). Here, we use a simpler technique to achieve the same goal, by using the standard KL-annealing strategy [28], [29] with less human intervention. Experimental results in Fig. 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 [30]. Thirdly, inspired by [23] we modify the cost function of UC-Net to reduce the discrepancy in encoding the latent variable at training and test time, which is elaborated in Section 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 [24] 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 [23] framework with an additional encoder to approximate the posterior distribution of z and an ABP [24] pipeline, which samples the latent variable directly from its true posterior distribution via Langevin dynamics based Markov chain Monte Carlo (MCMC) sampling [26]. 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 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) [23], [31], and Generative Adversarial Networks (GAN) [32], [33]. We also highlight the uniqueness of the proposed solutions in this section.

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 [1], [34], late-fusion models [18], [35] and cross-level fusion models [10]–[15], [17], [36]–[43]. 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. [34] 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. [35] introduced a late-fusion network (i.e., AFNet) to fuse predictions from the RGB and depth branch adaptively. In a similar pipeline, [18] 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, [36] presented a complementary-aware fusion block. [17] designed attention-aware cross-level combination blocks to obtain complementary information of each modality. [11] employed a fluid pyramid integration framework to achieve multi-scale cross-modal feature fusion. [13] designed a self-mutual attention model to effectively fuse RGB and depth information. Similarly, [12] presented a complimentary interaction module (CIM) to select complementary representation from the RGB and depth data. [14] provided joint learning and densely-cooperative fusion framework for complementary feature discovery. [15] 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 [44].

2.2 VAE or CVAE-based Deep Probabilistic Models

Ever since the seminal work by Kingma *et al.* [31] and Rezende *et al.* [45], VAE and its conditional counterpart CVAE [23] 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 [46], difference of motion modes [47], medical image segmentation models [48], and modeling inherent ambiguities of an image [49]. Meanwhile, VAE and CVAE have been explored in more complex vision tasks such as uncertain future forecast [50], salient feature enhancement [51], human motion prediction [52], [53], and shape-guided image generation [54]. Recently, VAE and CVAE have been extended to 3D domain targeting applications such as 3D meshes deformation [55], and point cloud instance segmentation [56]. For saliency detection, [57] adopted VAE to model image background, and separated salient objects from the background through the reconstruction residuals.

2.3 GAN or CGAN-based Dense Models

GAN [32] and its conditional counterparts [33] 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 [58]–[62] and treat the discriminator loss as a higher-order regularizer for dense prediction; or 2) apply GANs to 'semi-supervised scenarios [63], [64], 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, [65] introduced a discriminator in the fixation prediction network to distinguish predicted fixation map and ground-truth. Different from the above two directions, [66] adopted GAN in a RGB-D saliency detection network to explore the intra-modality (RGB, depth) and cross-modality simultaneously. [67] used GAN as a denoising technique to clear up the noisy input images. [62] designed a discriminator to
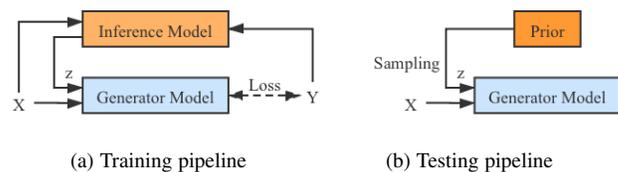

(a) Training pipeline (b) Testing pipeline

Fig. 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.

distinguish real saliency map (group truth) and fake saliency map (prediction), thus structural information can be learned without CRF [68] as post-processing technique. [69] adopted CycleGAN [70] as an domain adaption technique to generate pseudo-NIR image for existing RGB saliency dataset and achieve multi-spectral image salient object detection.

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 [1]. As a conditional latent variable model, two different solutions can be used to infer the latent variable. One is CVAE-based [23] method (the one we used in the preliminary version [1]), 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 [24].

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" [27], 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 [1] as an effective way to avoid posterior collapse problem. In this extended version, we propose a much simpler strategy using the KL annealing strategy [28], [29], 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 [24] 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 [1]. 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 3.2.
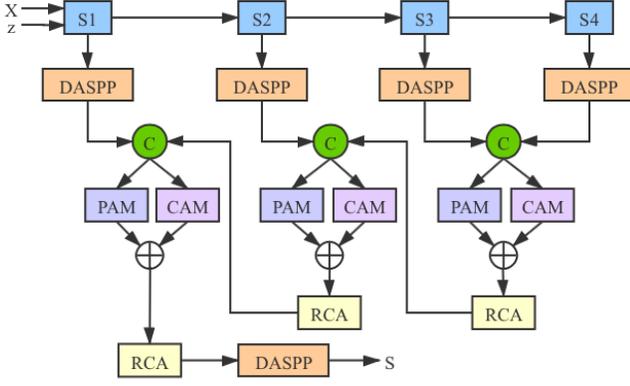
Fig. 3. Details of the "Generator Model", which takes image X and latent variable z as input, and produce stochastic saliency map S, where "S1-S4" represent the four convolutional blocks of our backbone network. "DASPP" is the DenseASPP module [71], "PAM" and "CAM" are position attention and channel attention module [30], "RCA" is the Residual Channel Attention operation from [72].

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 (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.

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 [73] as our backbone, which contains four convolutional blocks. To enlarge the receptive field, we follow DenseASPP [71] 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 [72] to obtain stochastic saliency map S. As illustrated in Fig. 3, our generator model follows the recent progress in dense prediction problems such as semantic segmentation [22], via a proper use of a hybrid attention mechanism. To this end, our generator model benefits from two

types of attention: a Position Attention Module [30] and a Channel Attention Module [30]. 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 [30] to aggregate and fuse the outputs of these two attention modules to further enhance the feature representations.

3.2 Inference Model

We propose two different solutions to infer or update the latent variable z: 1) A CVAE-based [23] pipeline, in which we approximate the posterior distribution via a neural network (*i.e.*, the encoder); and 2) An ABP [24] based strategy to sample directly from the true posterior distribution of z via Langevin Dynamics based MCMC [25].

Infer z with CVAE: The Variational Auto-encoder [31] 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:

$$\begin{aligned}\mathcal{L}_{\text{VAE}} = E_{z \sim Q_\phi(z|X)}[-\log P_\omega(X|z)] \\ + D_{KL}(Q_\phi(z|X)||P(z)),\end{aligned} \quad (1)$$

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 $D_{KL}(Q_\phi(z|X)||P(z))$ 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 [23] 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:

$$\begin{aligned}\mathcal{L}_{\text{CVAE}} = E_{z \sim Q_\phi(z|X,Y)}[-\log P_\omega(Y|X, z)] \\ + \lambda_{kl} * D_{KL}(Q_\phi(z|X, Y)||P_\theta(z|X)),\end{aligned} \quad (2)$$

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 2.4, we introduce a linear KL annealing [28], [29] 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
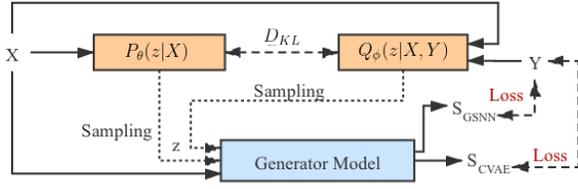
Fig. 4. RGB-D saliency detection via CAVE. The "Generator Model" is shown in Fig. 3. During training, we sample from both posterior net $z \sim Q_\phi(z|X,Y)$ and prior net $z \sim P_\theta(z|X)$ to obtain predictions S_{CVAE} and S_{GSNN} respectively. During testing, S_{GSNN} is our prediction.

Fig. 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.

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 [23], the conditional auto-encoding of output variables at training 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) [23], and the objective function is:

$$\mathcal{L}_{\text{GSNN}} = E_{z \sim P_\theta(z|X)}[-\log P_\omega(Y|X,z)]. \quad (3)$$

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

$$\mathcal{L}_{\text{Hybrid}} = \alpha\mathcal{L}_{\text{CVAE}} + (1-\alpha)\mathcal{L}_{\text{GSNN}} \quad (4)$$

Following the standard practice of CVAE [23], we design a CVAE-based RGB-D saliency detection pipeline as shown in Fig. 4. The two inference models ($Q_\phi(z|X,Y)$ and $P_\theta(z|X)$) share same structure as shown in Fig. 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 reparameterization trick: $z = \mu + \sigma \cdot \epsilon$, where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$.

According to Eq. 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. 4 can achieve structured outputs with hyper-parameter α to balance the two objective functions in Eq. 2 and Eq. 3.
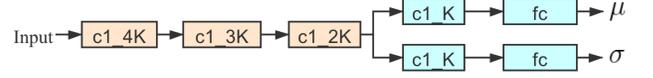
Algorithm 1 Learning Stochastic Saliency via Alternating Back-propagation

---

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 [73] 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. 8, 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**

Infer z with ABP: As mentioned earlier, one drawback of CVAE-based models is the posterior collapse problem [27], 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 [24]. 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 [24] 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" [24]. Secondly, given the updated latent variable, the network parameter set is updated with gradient descent, and they call it "Learning back-propagation" [24]. 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:

$$z \sim P(z) = \mathcal{N}(0, \mathbf{I}), \quad (5)$$

$$Y = f_\omega(X, z) + \epsilon, \epsilon \sim \mathcal{N}(0, \text{diag}(\sigma)^2), \quad (6)$$
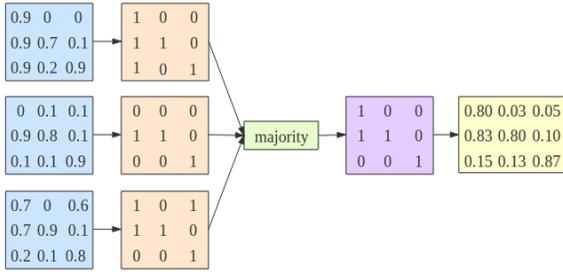
Fig. 6. Example showing how the saliency consensus module works.

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:

$$
\begin{aligned}
\frac{\partial}{\partial \omega} \log P_\omega(Y|X) &= \frac{1}{P_\omega(Y|X)} \frac{\partial}{\partial \omega} P_\omega(Y|X) \\
&= \mathrm{E}_{P_\omega(z|X,Y)} \left[\frac{\partial}{\partial \omega} \log P_\omega(Y,z|X) \right].
\end{aligned}
\tag{7}
$$

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 [24], we use Langevin Dynamics based MCMC (a gradient-based Monte Carlo method) to sample z, which iterates:

$$
z_{t+1} = z_t + \frac{s^2}{2} \left[\frac{\partial}{\partial z} \log P_\omega(Y, z_t|X) \right] + s\mathcal{N}(0, I_d), \tag{8}
$$

with

$$
\frac{\partial}{\partial z} \log P_\omega(Y, z|X) = \frac{1}{\sigma^2}(Y - f_\omega(X,z))\frac{\partial}{\partial z}f_\omega(X,z) - z, \tag{9}
$$

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 [23] inference model and ABP-based [24] 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 [27], where the latent variable z is independent of the prediction, making it unable to represent the uncertainty of labeling. To prevent posterior collapse, we adopt the KL annealing strategy [28], [29], and let the KL loss term in Eq. 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.

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 [23], 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 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 [16] 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 (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 (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 [74] 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 $P_{u,v}$, which is defined as:

$$
P_b^{mjv}(u, v) = \begin{cases} 1, & \sum_{c=1}^C P_b^c(u, v)/C \geq 0.5, \\ 0, & \sum_{c=1}^C P_b^c(u, v)/C < 0.5. \end{cases} \tag{10}
$$

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:

$$
P_g^{mjv}(u, v) = \frac{\sum_{c=1}^C (P_b^c(u, v) \times \mathbf{1}^c(u, v))}{\sum_{c=1}^C \mathbf{1}^c(u, v)}. \tag{11}
$$

We show one toy example with $C = 3$ in Fig. 6 to illustrate how the saliency consensus module works. As shown in Fig.

---

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.

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.

3.4 Loss function

We introduce two different inference models to update the latent variable z: a CVAE-based model as shown in Fig. 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 [75] to encourage disparities to be locally smooth with an L1 penalty on the disparity gradients. It was then adopted in [76] 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 [76], we define first-order derivatives of the saliency map in the smoothness term as

$$\mathcal{L}_{\text{Smooth}} = \sum_{u,v} \sum_{d \in \overrightarrow{x}, \overrightarrow{y}} \Psi(|\partial_d P_{u,v}| e^{-\alpha |\partial_d Ig(u,v)|}), \quad (12)$$

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 [76].

We need to compute intensity Ig of the image in the smoothness loss, as shown in Eq. (12). To achieve this, we follow a saliency-preserving [77] color image transformation strategy and convert the RGB image I to a gray-scale intensity image Ig as:

$$Ig = 0.2126 \times I^{lr} + 0.7152 \times I^{lg} + 0.0722 \times I^{lb}, \quad (13)$$

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:

$$I^{lr} = \begin{cases} \dfrac{I^r}{12.92}, & I^r \leq 0.04045, \\ \left(\dfrac{I^r + 0.055}{1.055}\right)^{2.4}, & I^r > 0.04045, \end{cases} \quad (14)$$

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. (14).

CVAE Inference Model based Loss Function: For the CVAE-based inference model, we show its loss function in Eq. 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 [7]. The structure-aware loss is a weighted extension of cross-entropy

loss, which integrates the boundary IOU loss [78] 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:

$$\mathcal{L}_{\text{sal}}^{CVAE} = \mathcal{L}_{\text{Hybrid}} + \lambda_1 \mathcal{L}_{\text{Smooth}}. \quad (15)$$

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:

$$\mathcal{L}_{ABP} = -\sum_{i=1}^{n} \log P_\omega(Y_i | X_i), \quad (16)$$

which can be the same structure-aware loss as in [7] 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:

$$\mathcal{L}_{\text{sal}}^{ABP} = \mathcal{L}_{ABP} + \lambda_2 \mathcal{L}_{\text{Smooth}}. \quad (17)$$

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

4 EXPERIMENTAL RESULTS

4.1 Setup

Datasets: We perform experiments on six datasets including five widely used RGB-D saliency detection datasets (namely NJU2K [85], NLPR [80], SSB [90], LFSD [91], DES [82]) and one newly released dataset (SIP [16]).

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) [98] and 4) mean Enhanced alignment measure (E-measure, E_ξ) [79].

- **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:

$$\text{MAE} = \frac{1}{N} |Sal - G|, \quad (18)$$

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 [98]. Thus, we additionally include the structure measure (S-measure [98]). The S-measure combines the region-aware (S_r) and object-aware (S_o) structural similarity as their final structure metric:

$$S_\alpha = \alpha * S_o + (1 - \alpha) * S_r, \quad (19)$$

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

- **E-measure E_ξ:** E-measure is the recent proposed Enhanced alignment measure [79] in the binary map evaluation field. This measure is based on cognitive vision

TABLE 1
Benchmarking results of ten 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 [79]. Evaluation tool: https://github.com/DengPingFan/D3NetBenchmark.

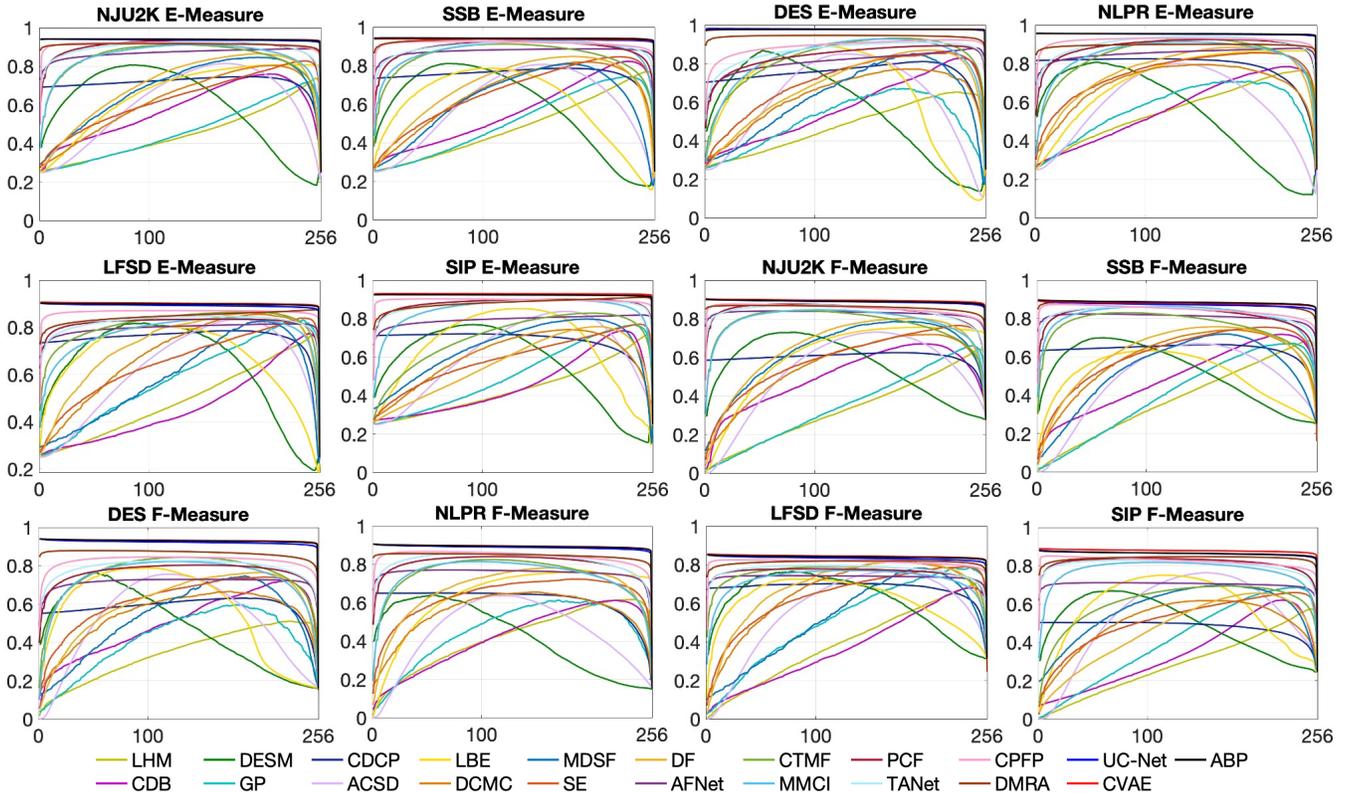|	Metric	Handcrafted Feature based Models										Deep Models								Ours		
		LHM [80]	CDB [81]	DESM [82]	GP [83]	CDCP [84]	ACSD [85]	LBE [86]	DCMC [87]	MDSF [88]	SE [89]	DF [34]	AFNet [35]	CTMF [18]	MMCI [37]	PCF [36]	TANet [17]	CPFP [11]	DMRA [10]	UC-Net [1]	CVAE	ABP
NJU2K [85]	$S_\alpha \uparrow$	.514	.632	.665	.527	.669	.699	.695	.686	.748	.664	.763	.822	.849	.858	.877	.879	.878	.886	.897	**.902**	.900
	$F_\beta \uparrow$	.328	.498	.550	.357	.595	.512	.606	.556	.628	.583	.653	.827	.779	.793	.840	.841	.850	.873	.886	**.893**	.889
	$E_\xi \uparrow$	.447	.572	.590	.466	.706	.594	.655	.619	.677	.624	.700	.867	.846	.851	.895	.895	.910	.920	.930	**.937**	**.937**
	$\mathcal{M} \downarrow$	.205	.199	.283	.211	.180	.202	.153	.172	.157	.169	.140	.077	.085	.079	.059	.061	.053	.051	.043	**.039**	**.039**
SSB [90]	$S_\alpha \uparrow$	.562	.615	.642	.588	.713	.692	.660	.731	.728	.708	.757	.825	.848	.873	.875	.871	.879	.835	.903	.898	**.904**
	$F_\beta \uparrow$	.378	.489	.519	.405	.638	.478	.501	.590	.527	.611	.617	.806	.758	.813	.818	.828	.841	.837	.884	.878	**.886**
	$E_\xi \uparrow$	.484	.561	.579	.508	.751	.592	.601	.655	.614	.664	.692	.872	.841	.873	.887	.893	.911	.879	.938	.935	**.939**
	$\mathcal{M} \downarrow$	.172	.166	.295	.182	.149	.200	.250	.148	.176	.143	.141	.075	.086	.068	.064	.060	.051	.066	.039	.039	**.037**
DES [82]	$S_\alpha \uparrow$	.578	.645	.622	.636	.709	.728	.703	.707	.741	.741	.752	.770	.863	.848	.842	.858	.872	.900	.934	.937	**.940**
	$F_\beta \uparrow$	.345	.502	.483	.412	.585	.513	.576	.542	.523	.618	.604	.713	.756	.735	.765	.790	.824	.873	.919	**.929**	.928
	$E_\xi \uparrow$	.477	.572	.566	.503	.748	.613	.650	.631	.621	.706	.684	.809	.826	.825	.838	.863	.888	.933	.967	**.975**	**.975**
	$\mathcal{M} \downarrow$	.114	.100	.299	.168	.115	.169	.208	.111	.122	.090	.093	.068	.055	.065	.049	.046	.038	.030	.019	**.016**	**.016**
NLPR [80]	$S_\alpha \uparrow$	.630	.632	.572	.655	.727	.673	.762	.724	.805	.756	.806	.799	.860	.856	.874	.886	.888	.899	**.920**	.917	.919
	$F_\beta \uparrow$	.427	.421	.430	.451	.609	.429	.636	.542	.649	.624	.664	.755	.740	.737	.802	.819	.840	.865	.891	**.893**	.891
	$E_\xi \uparrow$	.560	.567	.542	.571	.782	.579	.719	.684	.745	.742	.757	.851	.840	.841	.887	.902	.918	.940	.951	**.952**	**.852**
	$\mathcal{M} \downarrow$	.108	.108	.312	.146	.112	.179	.081	.117	.095	.091	.079	.058	.056	.059	.044	.041	.036	.031	.025	.025	**.024**
LFSD [91]	$S_\alpha \uparrow$	.557	.520	.722	.640	.717	.734	.736	.753	.700	.698	.791	.738	.796	.787	.794	.801	.828	.847	.864	**.868**	.866
	$F_\beta \uparrow$	.396	.376	.612	.519	.680	.566	.612	.655	.521	.640	.679	.736	.756	.722	.761	.771	.811	.845	.855	.857	**.859**
	$E_\xi \uparrow$	.491	.465	.638	.584	.754	.625	.670	.682	.588	.653	.725	.796	.810	.775	.818	.821	.863	.893	.901	**.904**	.903
	$\mathcal{M} \downarrow$	.211	.218	.248	.183	.167	.188	.208	.155	.190	.167	.138	.134	.119	.132	.112	.111	.088	.075	.066	**.065**	**.065**
SIP [16]	$S_\alpha \uparrow$	.511	.557	.616	.588	.595	.732	.727	.683	.717	.628	.653	.720	.716	.833	.842	.835	.850	.806	.875	**.883**	.876
	$F_\beta \uparrow$	.287	.341	.496	.411	.482	.542	.572	.500	.568	.515	.465	.702	.608	.771	.814	.803	.821	.811	.867	**.877**	.863
	$E_\xi \uparrow$	.437	.455	.564	.511	.683	.614	.651	.598	.645	.592	.565	.793	.704	.845	.878	.870	.893	.844	.914	**.927**	.921
	$\mathcal{M} \downarrow$	.184	.192	.298	.173	.224	.172	.200	.186	.167	.164	.185	.118	.139	.086	.071	.075	.064	.085	.051	**.045**	.049

Fig. 7. E-measure and F-measure curves on six testing datasets (NJU2K, SSB, DES, NLPR, LFSD and SIP). Best viewed on screen.

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

TABLE 2
The code type and inference time of existing approaches. M = Matlab. Pt = PyTorch. Tf = Tensorflow.

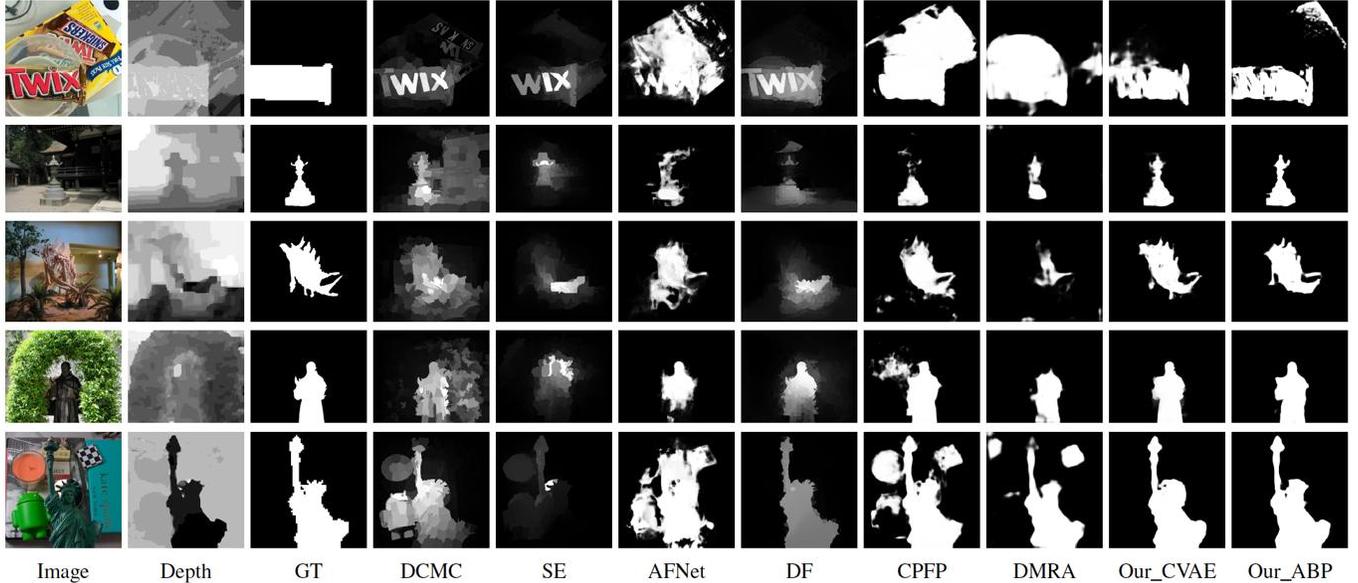| Method	LHM [80]	CDB [81]	DESM [82]	GP [83]	CDCP [84]	ACSD [92]	LBE [86]	DCMC [87]	MDSF [88]
Time (s)	2.13	0.60	7.79	12.98	60.00	0.72	3.11	1.20	60.00
Code Type	M	M	M	M&C++	M&C++	C++	M&C++	M	C++
Method	SE [89]	DF [34]	AFNet [35]	CTMF [18]	MMCI [37]	PCF [36]	CPFP [11]	Our_ABP	Our_CVAE
Time (s)	1.57	10.36	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

Fig. 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 3.3).

fixed (0-255) thresholds as shown in Fig. 7.

Implementation Details: We train our model using PyTorch, and initialized the encoder of the "Generator Model" with ResNet50 [73] parameters pre-trained on ImageNet. Inside the "DASPP" module of the "Generator Model" in Fig. 3, we use four different scales of dilation rate: 6, 12, 18, 24 same as [71], 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.

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 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 1 demonstrate the benefits of both CVAE and ABP which consistently achieve the best performance on all datasets. Specifically, on SSB [90] and SIP [16], 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 1. Moreover, compared with our preliminary version "UC-Net" [1], 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. 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 3, which further illustrates the superior performance of the proposed framework.

Qualitative Comparisons: In Fig. 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 [35], CPFP [11] and DMRA [10]) 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,

---

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 3
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 [93]	NLDF [78]	PiCANet [94]	RAS [95]	DGRL [96]	CPD [97]	SCRN [9]	F3Net [7]	CAVE Ours
NJU2K [92]	$S_\alpha \uparrow$	.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 [90]	$S_\alpha \uparrow$	.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 [82]	$S_\alpha \uparrow$	.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 [80]	$S_\alpha \uparrow$	.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 [91]	$S_\alpha \uparrow$	.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 [16]	$S_\alpha \uparrow$	.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**

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 [87], SE [89], AFNet [35], CPFP [11] 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. 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 [99], we first generate $B = 100$ easy and difficult samples. To achieve this, we first adopt three different conventional saliency models (RBD [100], MR [101] and GS [102], which rank among the top six conventional hand-crafted feature based RGB saliency models [74]), 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

4. We use the RGB data only.

and B images with the largest entropy as the 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. 10.

Inference Time[5] Comparison: We summarize basic information of competing methods in Table 2 for clear comparison, including their code type and inference time. Table 2 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.

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. 9. The "Our_CVAE Samples" in Fig. 9 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. 9 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 "Our_ABP") is consistent with the provided GT, which verifies effectiveness of both our latent variable and the "Saliency Consensus Module".

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 [103] in particular), and the solution to prevent network from posterior collapse. We show the performance in Table 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. 11 (a). We further design a middle fusion models and a late fusion model as shown in Fig. 11 (b) and (c) respectively. The performance of each model is shown in Table 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 [27], inspired by [52], 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,

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.

TABLE 4
Evaluation of the effect of different components in our models, and alternative structures. We present mean F_β and mean E_ξ.

Method	NJU2K $S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$M \downarrow$	SSB $S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$M \downarrow$	DES $S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$M \downarrow$	NLPR $S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$M \downarrow$	LFSD $S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$M \downarrow$	SIP $S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$M \downarrow$
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

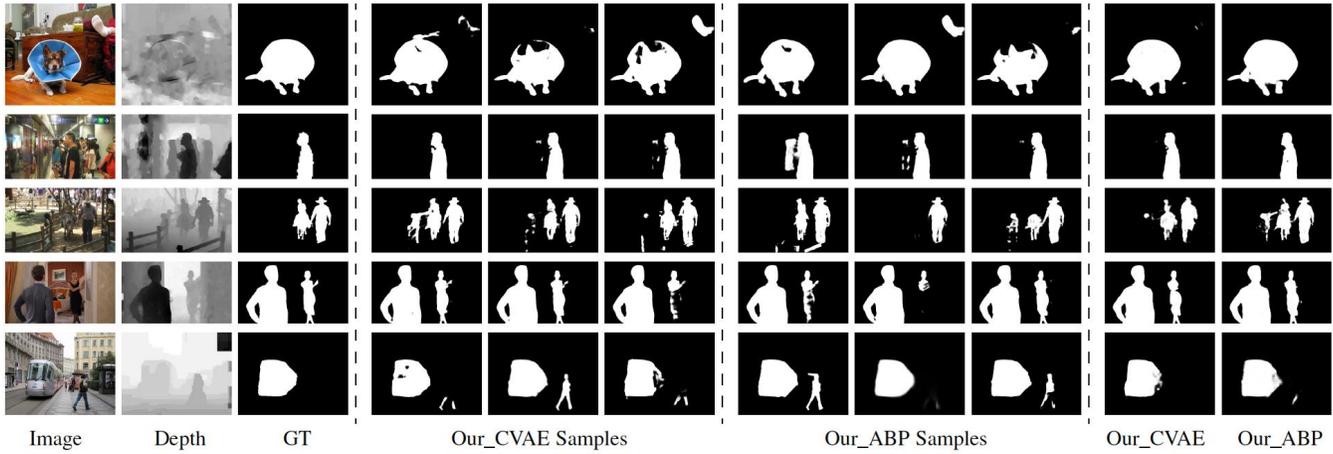Fig. 9. Structured outputs generation, where "Our_CVAE Samples" and "Our_CVAE" are samples and the deterministic prediction respectively.

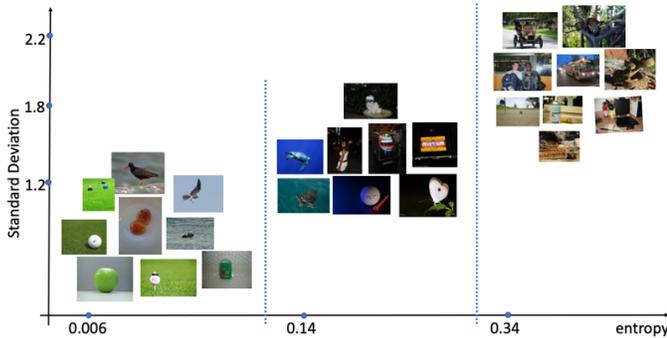Image   Depth   GT   Our_CVAE Samples   Our_ABP Samples   Our_CVAE   Our_ABP

Fig. 10. Image distribution by analysing entropy and standard deviation.

and replace S4 with S4' in Fig. 3. For the late fusion model, the "Generator Model" represents the generator model in Fig. 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".

Analysing the Effect of the Dimension of z: The scale of z may influence both network performance and diversity of predictions. In this paper, 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. 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. Meanwhile, 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 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 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

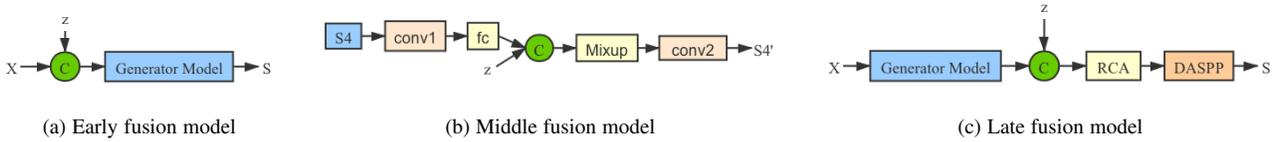(a) Early fusion model　　　　　　　　　　(b) Middle fusion model　　　　　　　　　　(c) Late fusion model

Fig. 11. Detail network structures of different fusion schemes: the early fusion model (a), the middle fusion model (b) and the late fusion model (c).
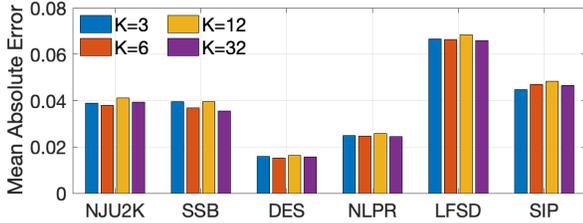

Fig. 12. Dimension analysis of the latent variable.

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. 4. To test how our network performs with only the CVAE loss in Eq. 2 or GSNN loss in Eq. 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. Meanwhile, 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. 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 [7], 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 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 [103] 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 pixels local surface normal makes with the inferred gravity direction. HHA is widely used in RGB-D dense models [18], [104] 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 [105].

Training without KL Annealing: As discussed in Section 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 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. 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. 13. Meanwhile, we show the mean variance of our CVAE-based and ABP-based models as "CVAE" and "ABP" respectively. Fig. 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".

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 5. We train our model ("Ours_CVAE" and "Ours_ABP") with DUTS training dataset [109], and evaluate performance of our methods and competing methods on six widely-used benchmarks: (1) DUTS testing dataset; (2) ECSSD [110]; (3) DUT [101]; (4) HKU-IS [111]; (5) THUR [112] and (6) SOC [113]. 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.
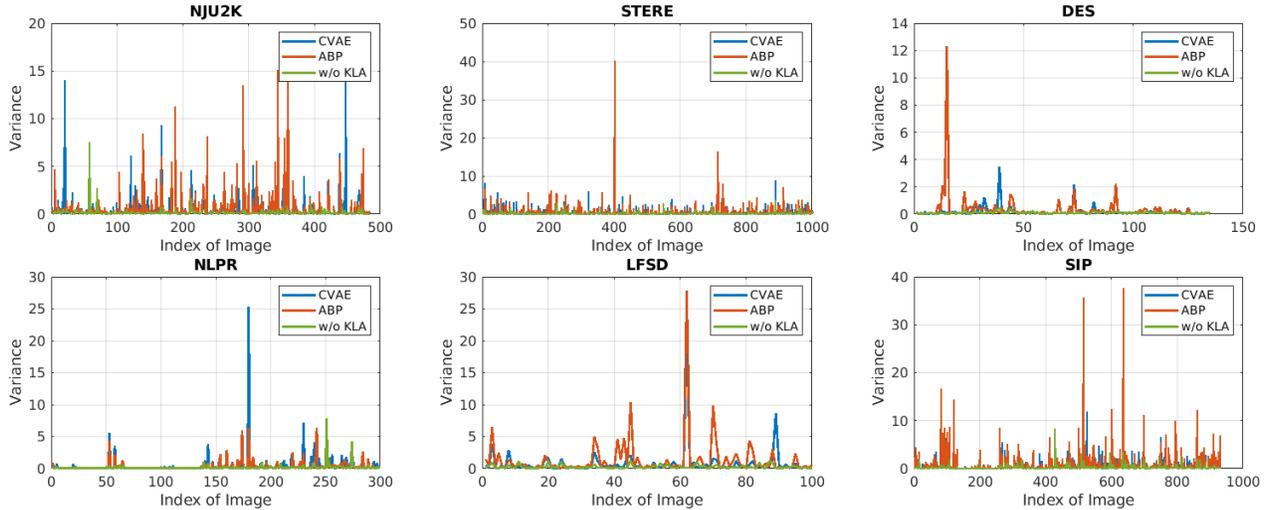
Fig. 13. Mean variance of multiple predictions using our CAVE-based model ("CVAE"), ABP-based model ("ABP"), and the CAVE-based model without KL annealing term ("w/o KLA"). Best viewed on screen.

TABLE 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.

Method	DUTS				ECSSD				DUT				HKU-IS				THUR				SOC			
	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$
DGRL [96]	.846	.790	.887	.051	.902	.898	.934	.045	.809	.726	.845	.063	.897	.884	.939	.037	.816	.727	.838	.077	-	-	-	-
PiCAN [94]	.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 [78]	.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 [106]	.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 [93]	.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 [107]	.862	.792	.883	.049	.905	.886	.922	.048	.809	.710	.831	.064	.907	.878	.930	.039	.819	.718	.829	.079	-	-	-	-
SCRN [9]	.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 [108]	**.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

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. Meanwhile, 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 the similar pipeline as 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.

Meanwhile, 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. Furthermore, we extend our solutions to 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.

Two different inference models are introduced to learn the proposed generative network as shown in Fig. 2 (a). From our experience, both the CVAE-based and ABP-based inference models can lead to diverse saliency predictions as shown in Fig. 13. While, as extra encoder 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 several steps of Langevin Dynamics based MCMC as shown in Eq. 8, which leads to relatively longer training time.

In the future, we would like to extend our approach to other saliency detection problems. Also, we plan to capture new datasets with multiple human annotations to further model the statistics of human uncertainty in saliency perception.

REFERENCES

[1] 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.

[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.

[3] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 1597–1604, 2009.

[4] 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.

[5] 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.*, pp. 9029–9038, 2018.

[6] Y. Liu, Q. Zhang, D. Zhang, and J. Han, "Employing deep part-object relationships for salient object detection," in *Int. Conf. Comput. Vis.*, 2019.

[7] S. W. Jun Wei and Q. Huang, "F3net: Fusion, feedback and focus for salient object detection," in *AAAI Conf. Art. Intell.*, 2020.

[8] 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.

[9] Z. Wu, L. Su, and Q. Huang, "Stacked cross refinement network for edge-aware salient object detection," in *Int. Conf. Comput. Vis.*, 2019.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] H. Chen and Y. Li, "Three-stream attention-aware network for RGB-D salient object detection," *IEEE T. Image Process.*, pp. 2825–2835, 2019.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 3431–3440, 2015.

[23] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Adv. Neural Inform. Process. Syst.*, pp. 3483–3491, 2015.

[24] T. Han, Y. Lu, S. Zhu, and Y. Wu, "Alternating back-propagation for generator network," in *AAAI Conf. Art. Intell.*, 02 2017.

[25] R. M. Neal, *MCMC Using Hamiltonian Dynamics*, vol. 54, pp. 113–162. CRC Press, 2011.

[26] R. M. Neal, "MCMC using hamiltonian dynamics," *Handbook of Markov Chain Monte Carlo*, vol. 54, pp. 113–162, 2010.

[27] 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.

[28] 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.), pp. 3738–3746, 2016.

[29] 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.

[30] 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.*, pp. 3146–3154, 2019.

[31] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Int. Conf. Learn. Represent.*, 2013.

[32] 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.*, pp. 2672–2680, 2014.

[33] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *CoRR*, vol. abs/1411.1784, 2014.

[34] 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.

[35] N. Wang and X. Gong, "Adaptive fusion for RGB-D salient object detection," *arXiv:1901.01369*, 2019.

[36] H. Chen and Y. Li, "Progressively complementarity-aware fusion network for RGB-D salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 3051–3060, 2018.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] 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.

[44] 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.

[45] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Int. Conf. Mach. Learn.*, pp. 1278–1286, 2014.

[46] 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.

[47] 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.*, pp. 276–293, 2018.

[48] 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*, pp. 119–127, 2019.

[49] 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.*, pp. 6965–6975, 2018.

[50] 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.*, pp. 835–851, 2016.

[51] A. Abid and J. Y. Zou, "Contrastive variational autoencoder enhances salient features," *CoRR*, vol. abs/1902.04601, 2019.

[52] 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.

[53] 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.

[54] P. Esser, E. Sutter, and B. Ommer, "A variational u-net for conditional appearance and shape generation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 8857–8865, 2018.

[55] 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.

[56] 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.

[57] B. Li, Z. Sun, and Y. Guo, "Supervae: Superpixelwise variational autoencoder for salient object detection," in *AAAI Conf. Art. Intell.*, pp. 8569–8576, 2019.

[58] P. Luc, C. Couprie, S. Chintala, and J. Verbeek, "Semantic segmentation using adversarial networks," in *Adv. Neural Inform. Process. Syst. Worksh.*

[59] 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)*, pp. 1–5, 2018.

[60] 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.

[61] H. Yu and X. Cai, "Saliency detection by conditional generative adversarial network," in *Ninth International Conference on Graphic and Image Processing*, p. 253, 04 2018.

[62] 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.

[63] 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.

[64] N. Souly, C. Spampinato, and M. Shah, "Semi supervised semantic segmentation using generative adversarial network," in *Int. Conf. Comput. Vis.*, pp. 5689–5697, 2017.

[65] 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.

[66] 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.

[67] 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.

[68] 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.), pp. 109–117, 2011.

[69] 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.*, pp. 12023–12030, 2020.

[70] 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.*, pp. 2242–2251, 2017.

[71] 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.*, pp. 3684–3692, 2018.

[72] 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.

[73] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 770–778, 2016.

[74] 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.

[75] C. Godard, O. M. Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 6602–6611, 2017.

[76] 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.

[77] 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.

[78] 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.*, pp. 6609–6617, 2017.

[79] 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.*, pp. 698–704, 2018.

[80] H. Peng, B. Li, W. Xiong, W. Hu, and R. Ji, "Rgbd salient object detection: a benchmark and algorithms," in *Eur. Conf. Comput. Vis.*, pp. 92–109, 2014.

[81] 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.

[82] Y. Cheng, H. Fu, X. Wei, J. Xiao, and X. Cao, "Depth enhanced saliency detection method," in *ACM ICIMCS*, pp. 23–27, 2014.

[83] 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.*, pp. 25–32, 2015.

[84] 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.

[85] 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.*, pp. 1115–1119, 2014.

[86] 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.*, pp. 2343–2350, 2016.

[87] 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.

[88] 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.

[89] J. Guo, T. Ren, and J. Bei, "Salient object detection for rgb-d image via saliency evolution," in *Int. Conf. Multimedia and Expo*, pp. 1–6, 2016.

[90] Y. Niu, Y. Geng, X. Li, and F. Liu, "Leveraging stereopsis for saliency analysis," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 454–461, 2012.

[91] N. Li, J. Ye, Y. Ji, H. Ling, and J. Yu, "Saliency detection on light field," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 2806–2813, 2014.

[92] 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.*, pp. 1115–1119, 2014.

[93] M. Feng, H. Lu, and E. Ding, "Attentive feedback network for boundary-aware salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[94] N. Liu, J. Han, and M.-H. Yang, "PiCANet: Learning Pixel-wise Contextual Attention for Saliency Detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 3089–3098, 2018.

[95] S. Chen, X. Tan, B. Wang, and X. Hu, "Reverse attention for salient object detection," in *Eur. Conf. Comput. Vis.*, 2018.

[96] 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.*, pp. 3127–3135, 2018.

[97] 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.

[98] 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.*, pp. 4548–4557, 2017.

[99] B. Settles, "Active learning literature survey," tech. rep., University of Wisconsin-Madison Department of Computer Sciences, 2009.

[100] W. Zhu, S. Liang, Y. Wei, and J. Sun, "Saliency optimization from robust background detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 2814–2821, 2014.

[101] 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.*, pp. 3166–3173, 2013.

[102] Y. Wei, F. Wen, W. Zhu, and J. Sun, "Geodesic saliency using background priors," in *Eur. Conf. Comput. Vis.*, pp. 29–42, 2012.

[103] 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.*, pp. 345–360, 2014.

[104] 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.*, pp. 11836–11845, 2019.

[105] C. M. Bishop, "Training with noise is equivalent to tikhonov regularization," *Neural Computation*, vol. 7, no. 1, pp. 108–116, 1995.

[106] 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.

[107] 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.

[108] 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.

[109] 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.*, pp. 136–145, 2017.

[110] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 1155–1162, 2013.

[111] G. Li and Y. Yu, "Visual saliency based on multiscale deep features," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 5455–5463, 2015.

[112] 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.

[113] 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.*, pp. 186–202, 2018.