Learning Generative Vision Transformer with Energy-Based Latent Space for Saliency Prediction

Anonymous Author(s)
Affiliation
Address
email

Abstract

Vision transformer networks have shown superiority in many computer vision tasks. In this paper, we take a step further by conducting research on generative vision transformer with latent variables following an informative energy-based prior for salient object detection. Both the vision transformer network and the energy-based prior model are jointly trained via Markov chain Monte Carlo (MCMC)-based maximum likelihood estimation, in which the sampling from the intractable posterior and prior distributions of the latent variables are performed by Langevin dynamics. Further, with the generative transformer, we can easily obtain pixel-wise uncertainty map indicating the model confidence of the prediction. Different from existing generative models which define the prior of the latent variable as a simple isotropic Gaussian distribution, our energy-based informative prior can be more expressive to capture the latent space of the data. We apply the proposed generative vision transformer to both RGB image and RGB-D image salient object detection tasks with a unified framework. Extensive experimental results illustrate that our solution can achieve both high performance model and meaningful confidence map representing awareness of model prediction.

1 Introduction

Salient object detection [70, 71, 17, 18, 5] has achieved significant performance improvement with the deep convolutional neural network revolution. A conventional framework for deep salient object detection involves two main modules: 1) a convolutional encoder-decoder for saliency map generation; 2) a loss function to guide the encoder-decoder for parameter updating. The goal of the above two modules is to achieve a mapping function from the input image space to the output saliency space.

Two main issues exist in the above conventional deep saliency frameworks: 1) the sliding window based convolution operation makes the deep saliency model less effective in modeling the global contrast, which is essential for salient object detection [8]; 2) the one-to-one deterministic mapping mechanism makes the current framework impossible to estimate the pixel-wise confidence [21] of model prediction. Although [83] introduces a conditional variational autoencoder (CVAE) [62] for RGB-D salient object detection, the potential posterior collapse problem [24] makes the stochastic predictions less effective in generating meaningful uncertainty estimation.

Motivated by above two issues, we conduct research on generative vision transformer for salient object detection, where a vision transformer network [45] is chosen as backbone to design our generative framework. On the one hand, Transformer [65] is proven to be effective in long-range dependency modeling, which is capable of modeling various scopes of object context information with the multi-head self-attention module. With such a architecture, we can achieve global context modeling for effective salient object detection. On the other hand, as a generative model, the randomness of the latent variables enables our model to produce stochastic predictions for uncertainty estimation.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

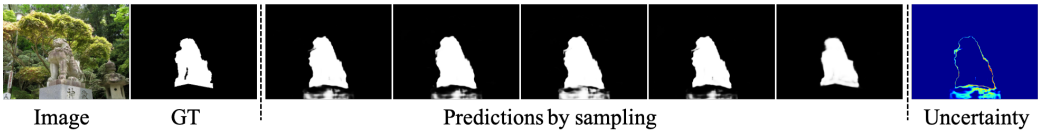

Figure 1: Outputs of our generative vision transformer based salient object detection network, including stochastic predictions ("Predictions by sampling") and the estimated uncertainty map ("Uncertainty").

Nowadays, there are two types of generative models that have been widely studied, namely the variational autoencoder (VAE) [31] and the generative adversarial network (GAN) [19]. To train a top-down latent variable generator, the former introduces an extra encoder to approximate the posterior distribution of the latent variables, and the latter designs a discriminator to distinguish the generated samples and the real data. [23, 74] present the third generative model, namely alternating back-propagation (ABP), with latent variables directly sampled from the true posterior distribution by using a gradient-based MCMC, e.g., Langevin dynamics [49]. All the three generative models define the prior of the latent variables as a simple none-informative isotropic Gaussian distribution, which is less expressive in capturing the true data distribution.

In this paper, we investigate generative modeling of the vision transformer. We construct a generative model for salient object detection in the form of top-down conditional latent variable model. Specifically, we propose a generative vision transformer by adding latent variables into the traditional deterministic transformer, and assume the latent variables follow an informative energy-based prior distribution. We parameterize the energy-based function by a trainable deep net. We jointly train the parameters of the EBM prior and the transformer network by maximum likelihood estimation (MLE). The MLE algorithm relies on MCMC sampling to evaluate the intractable prior and posterior distributions of the latent variables. Experimental results on RGB and RGB-D image salient object detection show that the generative framework equipped with the EBM prior and the transformer-based non-linear mapping is powerful in representing the conditional distribution of object saliency given an image, leading to more reasonable uncertainty estimation as shown in Fig. 1.

We summarize our main contributions as: 1) we propose a top-down generative vision transformer network with an energy-based prior distribution on latent space for salient object detection; 2) we propose to jointly train the vision transformer network and the energy-based prior model by MCMC-based maximum likelihood estimation; 3) We achieve new benchmark results for RGB and RGB-D image saliency detection, and meaningful uncertainty maps representing model's awareness of predictions.

2 Related Work

RGB salient object detection: The main focus of existing deep fully-supervised RGB image based SOD models [72, 46, 43, 73, 70, 66] is designing effective decoder for effective feature aggregation. [73] introduce extra edge module and uses the interaction between the edge module and the detection module to optimize the two tasks at the same time. [70] adaptively select complementary features while performing multi-scale feature aggregation. It also present a new structure-aware loss function for edge-aware saliency prediction. Similarly, [58] uses multiple supervisions and high-quality boundaries to guide the network to gradually optimize the saliency prediction.

RGB-D salient object detection: With both RGB image and depth data, one of the main focuses of the fully-supervised RGB-D image pair based salient object detection models is to explore the complementary information between the two modalities. [59, 83] present early-fusion models which concatenate the RGB image and depth data in the input layer. [68, 22, 57] are late-fusion models that perform salient object detection with each modal of data, and the fusion is achieved at the end of the network. The majority of RGB-D salient object detection models are cross-level fusion models [56, 5, 7, 6, 87, 84, 44, 17, 27, 54, 86, 36, 37], which perform feature-level fusion instead of the image level (early-fusion) or saliency map level (late-fusion) fusion.

Transformer and its applications: Transformer [65] has achieved great success in natural language processing (NLP). The breakthroughs of "Transformer" networks [13, 60, 45] in NLP domain has sparked great interest in the computer vision community to adapt these models for vision tasks such as object detection [4, 92, 11, 69, 81], image segmentation [89, 60, 69, 45], object tracking [77, 10, 78], pose estimation [47, 64], optical flow [28] *etc.*. DPT [60] adopts a U-shape structure

and uses ViT as encoder to encode features from different spatial resolutions of the initial embedding throughout all transformer stages. And a decoder assembles the set of tokens into image-like feature representations at various resolutions. Such feature representations are progressively fused into the final dense prediction, *e.g.*semantic segmentation and depth estimation. Swin [45] presents a hierarchical transformer with a shifted windowing scheme to achieve efficient transformer network with high resolution images as input.

Dense generative models: Two types of generative models have been widely studied, namely VAE [62, 31] and GAN [19]. VAE uses extra encoder to constrain the distribution of the latent variable, and GAN designs a discriminator to distinguish real samples and the generated samples. VAE [31, 62] has been applied to various tasks such as medical image segmentation [3] and ambiguities modeling [33]. For salient object detection, [35] adopted a VAE to model the image background, and separate salient objects from the background through reconstruction residuals. [83, 82] design CVAE based frameworks to model the subjective nature of saliency. The main scenarios of using GAN [19] can be roughly divided into two categories, namely fully-supervised and semi-supervised. The former [20, 34] treats the discriminator as learnable loss function to distinguish model prediction and the ground truth, and the latter [63, 26] uses GAN to explore the contribution of the unlabeled set. ABP [82] has also been used for RGB-D salient object detection to model dataset bias.

3 Generative Transformer with Energy-Based Latent Space

3.1 Model

We formulate the supervised saliency prediction problem as a conditional generative learning problem. Let $\mathbf{I} \in \mathbb{R}^D$ be an observed image, $s \in \mathbb{R}^D$ be the saliency map, and $z \in \mathbb{R}^d$ be the latent variables, where $D \gg d$. Consider the following generative model to predict a saliency map s from an image $\mathbf{I}$,

$$s = T_\theta(\mathbf{I}, z) + \epsilon, \quad z \sim p_\alpha(z), \quad \epsilon \sim \mathcal{N}(0, \sigma^2 I_D), \tag{1}$$

where T_θ is the non-linear mapping process from $[z, \mathbf{I}]$ to s with parameters θ, $p_\alpha(z)$ is the prior distribution with parameters α, and $\epsilon \sim \mathcal{N}(0, \sigma^2 I_D)$ is the observation residual. Due to the stochasticity of the latent variables z, given an image $\mathbf{I}$, its saliency map is also stochastic. Such a probabilistic model is in accord with the uncertainty of the image saliency.

The prior $p_\alpha(z)$ is not assumed to be a simple isotropic Gaussian distribution as GAN [19], VAE [31, 62] or ABP [23]. Specifically, it is in the form of the energy-based correction [53] of an isotropic Gaussian reference distribution $p_0(z)$ [76], i.e.,

$$p_\alpha(z) = \frac{1}{Z(\alpha)} \exp\left[-U_\alpha(z)\right] p_0(z) \propto \exp\left[-U_\alpha(z) - \frac{1}{2\sigma^2}||z||^2\right], \tag{2}$$

where $\mathcal{E}(z) = U_\alpha(z) + \frac{1}{2\sigma^2}||z||^2$ is the energy function that maps the latent variables z to a scalar, and $U_\alpha(z)$ is parameterized by a multi-layer perceptron (MLP) with parameters α. σ is a hyperparameter. $Z(\alpha) = \int \exp[-U_\alpha(z)]p_0(z)dz$ is the intractable normalizing constant that resolves the requirement for a probability distribution to have a total probability equal to one. $p_\alpha(z)$ is an informative prior in our model and needs to be estimated along with the non-linear mapping T_θ from the training data.

The mapping T_θ is parameterized by an image transformer [45] with self-attention mechanism, which encodes the image $\mathbf{I}$ and then decodes it along with the latent variables z to the saliency map s, thus, $p_\theta(s|\mathbf{I}, z) = \mathcal{N}(T_\theta(\mathbf{I}, z), \sigma^2 I_D)$.

3.2 Learning

The generative transformer with an energy-based prior, which is presented in Eq.(1), can be trained via maximum likelihood estimation. For notation simplicity, let $\beta = (\theta, \alpha)$. For the training examples $\{(\mathbf{I}_i, s_i), i = 1, ..., n\}$, the observed-data log-likelihood function is defined as

$$L(\beta) = \sum_{i=1}^{n} \log p_\beta(s_i|\mathbf{I}_i) = \sum_{i=1}^{n} \log \left[\int p_\beta(s_i, z_i|\mathbf{I}_i)dz\right] = \sum_{i=1}^{n} \log \left[\int p_\alpha(z_i)p_\theta(s_i|\mathbf{I}_i, z_i)dz\right]. \tag{3}$$

Maximizing $L(\beta)$ is equivalent to minimizing the Kullback-Leibler (KL) divergence between the model $p_\beta(s|\mathbf{I})$ and the data distribution $p_{\text{data}}(s|\mathbf{I})$. The gradient of $L(\beta)$ can be computed based on

$$\nabla_\beta \log p_\beta(s|\mathbf{I}) = \mathrm{E}_{p_\beta(z|s,\mathbf{I})}\left[\nabla_\beta \log p_\beta(s, z|\mathbf{I})\right] = \mathrm{E}_{p_\beta(z|s,\mathbf{I})}[\nabla_\beta(\log p_\alpha(z) + \log p_\theta(s|\mathbf{I}, z))], \tag{4}$$

where the posterior distribution $p_\beta(z|s,\mathbf{I}) = p_\beta(s,z|\mathbf{I})/p_\beta(s|\mathbf{I}) = p_\alpha(z)p_\theta(s|\mathbf{I},z)/p_\beta(s|\mathbf{I})$.

Algorithm 1 Maximum likelihood learning algorithm for generative vision transformer with energy-based latent space for saliency prediction

Input: (1) Training images $\{\mathbf{I}_i\}_i^n$ with associated saliency maps $\{s_i\}_i^n$; (2) Maximal number of learning iterations M; (3) Numbers of Langevin steps for prior and posterior $\{K_0, K_1\}$; (4) Langevin step sizes for prior and posterior $\{\delta_0, \delta_1\}$; (5) Learning rates for energy-based prior model and transformer $\{\xi_\alpha, \xi_\theta\}$.
Output: Parameters θ for the transformer and α for the energy-based prior model

1: Initialize θ and α
2: **for** $t \leftarrow 1$ to M **do**
3: Sample observed image-saliency pairs $\{(\mathbf{I}_i, s_i)\}_i^n$
4: For each $(\mathbf{I}_i, s_i)$, sample the prior $z_i^- \sim p_{\alpha_t}(z)$ using K_0 Langevin steps in Eq.(7) with a step size δ_0.
5: For each $(\mathbf{I}_i, s_i)$, sample the posterior $z_i^+ \sim p_{\beta_t}(z|s_i, \mathbf{I}_i)$ using K_1 Langevin steps in Eq.(8) with a step size δ_1.
6: Update energy-based prior by Adam with the gradient $\nabla\alpha$ computed in Eq.(9) and a learning rate ξ_α.
7: Update transformer by Adam with the gradient $\nabla\theta$ computed in Eq.(10) and a learning rate ξ_θ.
8: **end for**

Eq. (4) can be decomposed into two parts, i.e., the gradient for the energy-based model α

$$\mathrm{E}_{p_\beta(z|s,\mathbf{I})}[\nabla_\alpha \log p_\alpha(z)] = \mathrm{E}_{p_\alpha(z)}[\nabla_\alpha U_\alpha(z)] - \mathrm{E}_{p_\beta(z|s,\mathbf{I})}[\nabla_\alpha U_\alpha(z)], \tag{5}$$

and the gradient for the transformer θ

$$\mathrm{E}_{p_\beta(z|s,\mathbf{I})}[\nabla_\theta \log p_\theta(s|\mathbf{I},z)] = \mathrm{E}_{p_\beta(z|s,\mathbf{I})}\left[\frac{1}{\sigma^2}(s - T_\theta(\mathbf{I},z))\nabla_\theta T_\theta(\mathbf{I},z)\right]. \tag{6}$$

$\nabla_\alpha U_\alpha(z)$ in Eq. (5) and $\nabla_\theta T_\theta(\mathbf{I},z)$ in Eq. (6) can be efficiently computed via back-propagation. Both Eq. (5) and Eq. (6) include intractable expectation terms $\mathrm{E}_p(\cdot)$, which can be approximated by Markov chain Monte Carlo (MCMC) [42] samples. To be specific, we can use a gradient-based MCMC, e.g., Langevin dynamics [49], which is initialized with a Gaussian noise distribution p_0, to draw samples from the energy-based prior model $p_\alpha(z)$ by iterating

$$z_{t+1} = z_t - \delta\nabla_z U_\alpha(z_t) + \sqrt{2\delta}\epsilon_t, \quad z_0 \sim p_0(z), \epsilon_t \sim \mathcal{N}(0, I), \tag{7}$$

and draw samples from the posterior distribution $p_\beta(z|s,\mathbf{I})$ by iterating

$$z_{t+1} = z_t - \delta\left[\nabla_z U_\alpha(z) - \frac{1}{\sigma^2}(s - T_\theta(\mathbf{I},z_t))\nabla_z T_\theta(\mathbf{I},z_t)\right] + \sqrt{2\delta}\epsilon_t, \quad z_0 \sim p_0(z), \epsilon_t \sim \mathcal{N}(0, I). \tag{8}$$

δ is the Langevin step size and can be specified independently in Eq. (7) and Eq. (8). We use $\{z_i^+\}$ and $\{z_i^-\}$ to denote, respectively, the samples from the posterior distribution and the prior distribution. The gradients of α and θ can be computed with $\{\mathbf{I}_i\}$, $\{z_i^+\}$ and $\{z_i^-\}$ by

$$\nabla\alpha = \frac{1}{n}\sum_{i=1}^{n}[\nabla_\alpha U_\alpha(z_i^-)] - \frac{1}{n}\sum_{i=1}^{n}\left[\nabla_\alpha U_\alpha(z_i^+)\right], \tag{9}$$

$$\nabla\theta = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{1}{\sigma^2}(s_i - T_\theta(\mathbf{I}_i, z_i^+))\nabla_\theta T_\theta(\mathbf{I}_i, z_i^+)\right], \tag{10}$$

We can update the parameters with $\nabla\alpha$ and $\nabla\theta$ via the Adam optimizer [30]. We present the full learning and sampling algorithm of our model in Algorithm 1.

3.3 Network

Generative Vision Transformer: We design a generative vision transformer with the Swin transformer [45] backbone as shown in Fig. 2, which takes a three-channel image $\mathbf{I}$ and the latent variable z as input and output a one-channel saliency map $T_\theta(\mathbf{I},z)$. Two main modules are included in our generative vision transformer, including a "Transformer Encoder" and a "Feature Aggregation" module. The former takes $\mathbf{I}$ as input and produce a list of feature maps $\{f_l\}_{l=1}^5$ of channel size 128,256,512,1024 and 1024 respectively. The latter takes $\{f_l\}_{l=1}^5$ and the latent variable z as input to generate the saliency prediction $T_\theta(\{f_l\}_{l=1}^5, z)$.

Specifically, we first extend the latent variable z to same spatial size of f_5. Then we concatenate the extended z with f_5 and feed it to a 3×3 convolutional layer to produce feature map f_5' of same channel size as f_5, and define it as the new f_5. We then gradually concatenate higher level feature f_l with the lower level feature f_{l-1} and feed it to the channel attention module [85] $C_m([f_l, f_{l-1}])$ for discriminative feature aggregation, where C_m is the channel attention module, $[,]$ is the concatenation operation. We upsample f_l to the same spatial size as f_{l-1} before the concatenation. $C_m([f_l, f_{l-1}])$ is then mapped to same channel size as f_{l-2} with a 3×3 convolutional layer. We perform the feature concatenation until we reach f_1. Finally, we map the feature with another one 3×3 convolutional layer to obtain our one channel saliency map $T_\theta(\mathbf{I}, z)$.

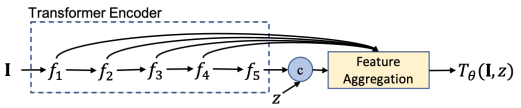

Figure 2: Generative latent variable vision transformer

Energy-based prior model: We design an energy-based model for the latent space by parameterzing the energy function via a multilayer perceptron (MLP), which uses four fully connected layers to map the latent variables z to the energy. The sizes of the feature maps of different layers of the MLP are $C_e, C_e/2, C_e/4, C_e$ respectively, where $C_e = 60$ in our experiment. "GELU" [25] activation is used after each fully connected layer except the last one.

3.4 Analysis

Convergence: Theoretically, when the Adam optimizer of $\beta = (\theta, \alpha)$ in the learning algorithm converges to a local minimum, it solve the following estimating equations

$$\nabla\theta = 0 \;\; \Rightarrow \;\; \frac{1}{n}\sum_{i=1}^{n} \mathrm{E}_{p_\beta(z|s_i,\mathbf{I}_i)}\left[\frac{1}{\sigma^2}(s_i - T_\theta(\mathbf{I}_i, z))\nabla_\theta T_\theta(\mathbf{I}_i, z)\right] = 0, \tag{11}$$

$$\nabla\alpha = 0 \;\; \Rightarrow \;\; \frac{1}{n}\sum_{i=1}^{n} \mathrm{E}_{p_\beta(z|s_i,\mathbf{I}_i)}[\nabla_\alpha U_\alpha(z)] - \mathrm{E}_{p_\alpha(z)}[\nabla_\alpha U_\alpha(z)] = 0, \tag{12}$$

which are the maximum likelihood estimating equations. However, in practise, the Langevin dynamics in Eq.(9) and Eq.(10) might not converge to the target distributions due to the use of a small number of Langevin steps (i.e., short-run MCMC), the estimating equations in Eq.(11) and Eq.(12) will corresponds to a perturbation of the MLE estimating equation according to [50, 51]. The learning algorithm can be justified as a Robbins-Monro [61] algorithm, whose convergence is theoretically sound. Our model can also be trained with an extra encoder as an amortized inference network [32] for $p_\beta(z|s, \mathbf{I})$ and an extra generator as an amortized sampling network [75] for $p_\alpha(z)$. In this work, we prefer to keep our training algorithm simple in order to avoid extra efforts for the design of the auxiliary network architectures. We will study the joint training strategy in our future work.

Accuracy: Compared with the GAN-based generative framework, our model is a likelihood-based generative framework that will not suffer from mode collapse [2]. In comparison with VAE-based generative framework, whose training is also based on likelihood, our MCMC-based maximum likelihood learning algorithm will not encounter the posterior collapse issue that is caused by amortized inference. On the other hand, the variational inference typically relies on an extra inference network for efficient inference of the latent variables given an image and saliency pair, however, the approximate inference might not be able to take the full place of the posterior distribution in practise. To be specific, we use $q_\phi(z|s, \mathbf{I})$ to denote the tractable approximate inference network with parameters ϕ. The variational inference seeks to optimize $\min_\beta \min_\phi \mathrm{KL}(p_{\text{data}}(s|\mathbf{I})q_\phi(z|s, \mathbf{I})||p_\beta(z, s|\mathbf{I}))$, which can be further decomposed into $\min_\beta \min_\phi \mathrm{KL}(p_{\text{data}}(s|\mathbf{I})||p_\beta(s|\mathbf{I})) + \mathrm{KL}(q_\phi(z|s, \mathbf{I})||p_\beta(z|s, \mathbf{I}))$. That is, the variational inference maximizes the conditional data likelihood plus a KL-divergence between the approximate inference network and the posterior distribution. Only when the $\mathrm{KL}(q_\phi(z|s, \mathbf{I})||p_\beta(z|s, \mathbf{I})) \to 0$, the variational inference will lead to the MLE solution, which is exactly the objective of our model. However, there might exist a gap between them in practise due to the improper design of the inference network. Our learning algorithm is much simpler and more accurate.

Computational and memory costs: From the learning perspective, due to the iterative Langevin sampling for the posterior and prior distributions of latent variables, our model is more time-consuming than generative models with amortized inference, such as VAE, which is roughly 2.4 times faster than ours on RGB image salient object detection. However, without relying on an extra inference network, our framework is efficient in memory and friendly for network design.

Table 1: Performance comparison with benchmark RGB salient object detection models.

Method	DUTS [67]				ECSSD [79]				DUT [80]				HKU-IS [38]				PASCAL-S [40]				SOD [48]			
	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$
CPD [72]	.869	.821	.898	.043	.913	.909	.937	.040	.825	.742	.847	.056	.906	.892	.938	.034	.848	.819	.882	.071	.799	.779	.811	.088
SCRN [73]	.885	.833	.900	.040	.920	.910	.933	.041	.837	.749	.847	.056	.916	.894	.935	.034	.869	.833	.892	.063	.817	.790	.829	.087
PoolNet [41]	.887	.840	.910	.037	.919	.913	.938	.038	.831	.748	.848	.054	.919	.903	.945	.030	.865	.835	.896	.065	.820	.804	.834	.084
BASNet [58]	.876	.823	.896	.048	.910	.913	.938	.040	.836	.767	.865	.057	.909	.903	.943	.032	.838	.818	.879	.076	.798	.792	.827	.094
EGNet [88]	.878	.824	.898	.043	.914	.906	.933	.043	.840	.755	.855	.054	.917	.900	.943	.031	.852	.823	.881	.074	.824	.811	.843	.081
F3Net [70]	.888	.852	.920	.035	.919	.921	.943	.036	.839	.766	.864	.053	.917	.910	.952	.028	.861	.835	.898	.062	.824	.814	.850	.077
ITSD [90]	.886	.841	.917	.039	.920	.916	.943	.037	.842	.767	.867	.056	.921	.906	.950	.030	.860	.830	.894	.066	.836	.829	.867	.076
Ours	**.912**	**.891**	**.951**	**.025**	**.936**	**.940**	**.964**	**.025**	**.858**	**.802**	**.892**	**.044**	**.928**	**.926**	**.966**	**.023**	**.874**	**.876**	**.918**	**.053**	**.850**	**.855**	**.886**	**.064**

Table 2: Performance comparison with benchmark RGB-D salient object detection models.

Method	NJU2K [29]				SSB [52]				DES [9]				NLPR [55]				LFSD [39]				SIP [16]			
	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$
BBSNet [17]	.921	.902	.938	.035	.908	.883	.928	.041	.933	.910	.949	.021	.930	.896	.950	.023	.864	.843	.883	.072	.879	.868	.906	.055
BiaNet [86]	.915	.903	.934	.039	.904	.879	.926	.043	.931	.910	.948	.021	.925	.894	.948	.024	.845	.834	.871	.085	.883	.873	.913	.052
CoNet [27]	.911	.903	.944	.036	.896	.877	.939	.040	.906	.880	.939	.026	.900	.859	.937	.030	.842	.834	.886	.077	.868	.855	.915	.054
UCNet [83]	.897	.886	.930	.043	.903	.884	.938	.039	.934	.919	.967	.019	.920	.891	.951	.025	.864	.855	.901	.066	.875	.867	.914	.051
JLDCF [18]	.902	.885	.935	.041	.903	.873	.936	.040	.931	.907	.959	.021	.925	.894	.955	.022	.862	.848	.894	.070	.880	.873	.918	.049
Ours	**.932**	**.927**	**.959**	**.026**	**.921**	**.905**	**.953**	**.030**	**.947**	**.940**	**.979**	**.014**	**.938**	**.922**	**.966**	**.019**	**.889**	**.876**	**.920**	**.052**	**.907**	**.913**	**.943**	**.035**

Table 3: Contribution of channel attention module analysis.

Method	NJU2K [29]				SSB [52]				DES [9]				NLPR [55]				LFSD [39]				SIP [16]			
	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$
D_Base	.918	.905	.949	.033	.903	.877	.943	.038	.937	.917	.970	.018	.925	.898	.958	.022	.868	.860	.899	.066	.893	.891	.930	.043
D_CBase	.930	.924	.959	.026	.915	.895	.949	.032	.941	.929	.975	.015	.934	.912	.962	.020	.875	.863	.905	.063	.898	.900	.939	.040
Res50	.879	.855	.912	.054	.876	.841	.912	.054	.886	.854	.909	.033	.889	.845	.927	.037	.809	.782	.840	.104	.861	.836	.898	.063
Res50_C	.897	.875	.925	.046	.886	.853	.918	.049	.904	.874	.941	.030	.907	.868	.940	.032	.819	.789	.850	.096	.881	.856	.917	.051

4 Experimental Results

4.1 Setup

Datasets: For the RGB SOD, we train the models using DUTS training dataset [67], and test on six benchmark datasets, including the DUTS testing dataset, ECSSD [79], DUT [80], HKU-IS [38], PASCAL-S [40] and the SOD testing dataset [48]. For RGB-D SOD, we follow the conventional training setting, in which the training set is a combination of 1,485 images from the NJU2K dataset [29] and 700 images from the NLPR dataset [55]. We then test model performances on the NJU2K testing set, NLPR testing set LFSD [39], DES [9], SSB [52] and the SIP [16] testing set.

Evaluation Metrics: We adopt four evaluation metrics to measure the performance, including Mean Absolute Error $\mathcal{M}$, Mean F-measure (F_β), Mean E-measure (E_ξ) [15] and S-measure (S_α) [14].

Implementation Details: We train our model in Pytorch with the Swin transformer backbone [45] trained on ImageNet-1K [12], and other newly added layers are randomly initialized. Empirically we set dimension d of the latent variable z as $d = 16$. We resize all the images and ground truth to 384×384 to fit the swin transformer [45]. The maximum epoch is 50. The initial learning rates are 2.5×10^{-5} for all the three tasks in this paper. The whole training takes 9 hours with batch size 6 on one NVIDIA GTX 2080Ti GPUs for each model. During testing, our model can process 15 images/second.

4.2 Performance Comparison

We compare our unified salient object detection network with state-of-the-art RGB SOD models and RGB-D SOD models, and show model performance in Table 1 and Table 2 respectively. The consistent better performance of our models indicates effectiveness of our framework. Especially for PASCAL-S dataset [40], which contains more than 40% samples with large salient objets, and the significant performance gap between "Ours" and the existing solutions demonstrates effectiveness of our framework. We further show predictions of our RGB-D saliency model and BBSNet [17] in Fig. 3. With the effective global context modeling, our transformer backbone based framework is superior in detecting salient objects of various scales (6th and last column of Fig. 3).

4.3 Ablation Study

We analysis our framework in this section with the RGB-D saliency detection task.

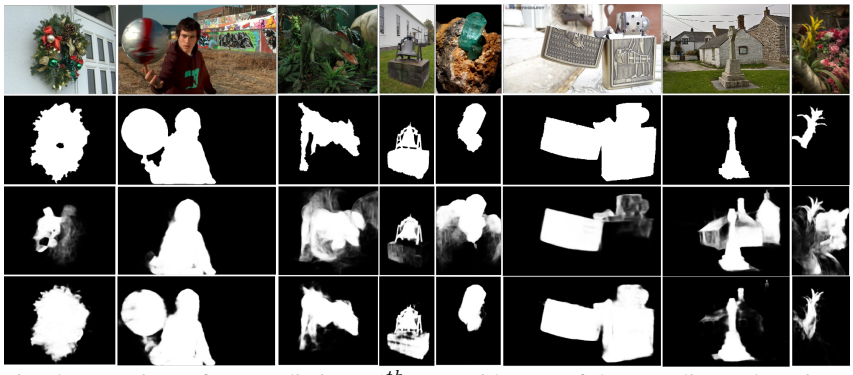

Figure 3: Visual comparison of our predictions ($4^{th} row$) with state-of-the-art saliency detection model (3rd row, "BBSNet" [17] in particular), where the 2nd row shows the ground truth saliency maps.

Table 4: Performance of other generative models with transformer backbone.

Method	NJU2K [29]				SSB [52]				DES [9]				NLPR [55]				LFSD [39]				SIP [16]			
	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$
G_Z	.930	.915	.951	.030	.918	.890	.942	.035	.945	.921	.969	.017	.940	.914	.962	.020	.883	.862	.902	.063	.907	.898	.935	.039
G_GAN	.928	.922	.956	.028	.916	.899	.950	.031	.945	.934	.979	.014	.934	.915	.961	.021	.889	.878	.921	.051	.901	.904	.941	.037
G_VAE	.928	.921	.955	.027	.914	.894	.947	.033	.942	.929	.975	.016	.934	.914	.961	.020	.874	.862	.907	.060	.904	.906	.942	.036
Ours	**.932**	**.927**	**.959**	**.026**	**.921**	**.905**	**.953**	**.030**	**.947**	**.940**	**.979**	**.014**	**.938**	**.922**	**.966**	**.019**	**.889**	**.876**	**.920**	**.052**	**.907**	**.913**	**.943**	**.035**

Backbone analysis: We argue that the long-range dependency modeling of the transformer backbone makes it quite suitable for the context based tasks, *e.g.*salient object detection. To test the superior performance of transformer backbone and conventional backbone, we design to deterministic learning framework with ResNet50 and Swin transformer encoder [45] as backbone respectively. The decoder part is same as our framework ("Feature Aggregation" in Fig. 2). Model performance is shown as "D_CBase" and "Res50_C" in Table. 3 respectively. We observe significant improvement of "D_CBase" over "Res50_C", which clearly indicates superior performance transformer backbone for salient object detection. Further, as claimed in [60, 45, 13] that the transformer backbone achieves larger receptive field, making it effective in detecting different scales of salient objects. We then visualize the performance of each model w.r.t.scale of the salient object on the SSB testing dataset [52], and show the result in Fig. 4. The relative stable performance of the transformer backbone w.r.t.different scales of the salient foreground indicates transformer backbone is scale-robust.

Generative model analysis: With the latent variable following the isotropic Gaussian distribution, we obtain an ABP [23] based model, a GAN [19] based model and a VAE [31, 62] based model shown as "G_Z", "G_GAN" and "G_VAE" respectively in Table 4. For "G_Z", we remove the EBM prior model, and keep the posterior Langevin part while training, and sample the latent variable directly from standard normal distribution during testing. For "G_GAN", we design a fully convolutional discriminator, which is composed of five 3×3 convolutional layers of stride 2 that map the input (concatenation of RGB image (or RGB-D image pair) with the prediction or ground truth) to features of channel size $k = 64, 64, 64, 64, 1$ gradually. The discriminator is trained to distinguish predictions (with an **0** as supervision) and ground truth saliency maps (with an **1** as supervision). For "G_VAE", we introduce extra latent feature encoder module to map the input image $\mathbf{I}$ to latent space of size d. The latent feature encoder consists of four convolutional layer of kernel size 4, stride 2, which maps the input image $\mathbf{I}$ to feature maps of channel size $c = 64, 64, 64, 64$. Then two fully connected layers are adopt to obtain the mean μ and standard deviation σ, which are then used to generate our VAE based latent variable $z = \mu + \sigma * \epsilon$ with the reparameteration trick, where ϵ is a random variable which follows the standard normal distribution. For all the three generative models, we define dimension of the latent variable same as our model, which is 16. We observe better or comparable performance of the three generative model based frameworks compared with our based model "D_CBase". Especially for LFSD [39] and SIP dataset [16], we found "G_GAN" achieves large performance improvement.

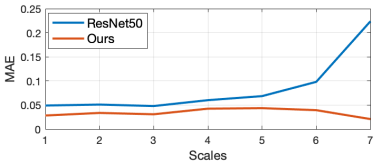

Figure 4: Scale robustness analysis.

Table 5: Hyper-parameter analysis.

Method	NJU2K [29]				SSB [52]				DES [9]				NLPR [55]				LFSD [39]				SIP [16]			
	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$
	Latent variable dimension																							
z8	.929	.916	.956	.027	**.929**	.897	.946	.032	.948	.923	.972	.015	.940	.915	.959	.020	**.889**	.865	.907	.061	.904	.899	.938	.035
z60	.925	.909	.956	.030	.915	.889	.950	.033	.948	.931	.976	.014	**.942**	.917	.963	**.019**	.880	.856	.906	.060	**.907**	.899	**.945**	**.034**
	EBM feature dimension																							
e20	.930	.913	.957	**.026**	.917	.898	.950	**.030**	**.951**	.921	.970	.016	.937	.913	.950	.022	.883	.860	.900	.063	.901	.892	.931	.037
e100	.926	.904	.950	.031	.916	.891	.952	.031	.942	.930	.970	.016	.939	.913	.960	.021	.875	.950	.895	.064	.903	.895	.934	.036
Ours	**.932**	**.927**	**.959**	**.026**	.921	**.905**	**.953**	**.030**	.947	**.940**	**.979**	**.014**	.938	**.922**	**.966**	**.019**	**.889**	**.876**	**.920**	**.052**	**.907**	**.913**	.943	.035

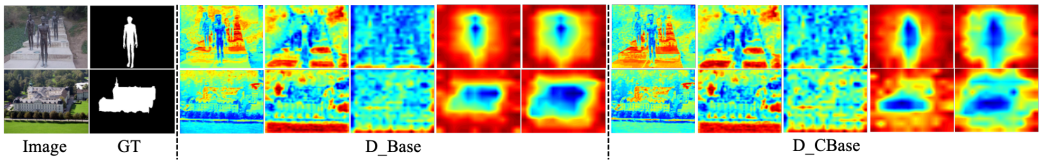

Figure 5: Backbone features of two different transformer based baseline models with ("D_CBase") and without ("D_Base") the residual channel attention module [85].

However, the performance gap between those generative models and "Ours" explain the superior performance of our framework of defining an EBM prior latent space.

Hyper-parameter analysis: We have two sets of hyper-parameters within our framework. The posterior Langevin related parameters as in Eq. 8 and the ebm prior langevin related parameters as in Eq. 7, including the latent variable dimension, EBM feature dimension (dimension of the last fully connected layer in the Energy-based prior model), langevin step size σ, langevin step number K_0 for prior langevin and K_1 for posterior Langevin. For the langevin step size and langevin step number, we find relative stable model performance with $\sigma = 0.4 \pm 0.2$, and for the langevin step number, we empirically set $K_0 = K_1 = 10$ to achieve trade-off between efficient training and model accuracy, as larger step number leads to longer time of training. Then we only analyse the latent variable dimension d and EBM feature dimension C_e and show performance in Table 5, where "z8" and "z60" indicate latent dimension $d = 8$ and $d = 60$ respectively. "e20" and "e100" represent the size of the last fully connected layer of EBM model $C_e = 20$ and $C_e = 100$ respectively. We observe relative stable performance with the above various hyper-parameter settings. The main reason lies in the high capacity of the generative vision transformer. We will further investigate model robustness w.r.t.more hyper-parameter settings in the future.

4.4 Discussion

We further analysis the transformer backbone within a generative framework.

Transformer encoder analysis: We introduce our generative vision transformer in Section 3.3, where extra residual channel attention module is used to extract channel-wise discriminative features. As a spatial attention modeling technique, we argue that the channel attention can serve as complementary attention modules for the transformer backbone based framework. To prove effectiveness of the channel attention module for salient object detection within the conventional backbone and transformer backbone, we remove the channel attention module [85] from both "D_CBase" and "Res50_C" in Table 3, leading to two new baseline model "D_Base" and "Res50" with two types of backbone as shown in Table 3. The significant performance decrease of "D_Base" (and "Res50") compared with "D_CBase" (and "Res50_C") indicates the importance of channel attention modules for both the conventional backbone and the transformer backbone. We further visualize the backbone feature of models with the channel attention module ("D_CBase") and without channel attention module ("D_Base") in Fig. 5, where we visualize different level of features of the transformer backbone. To achieve this, we compute channel-wise mean of each f_l, and then perform minmax normalization to generate the feature maps of different level of the backbone. We observe similar lower level feature maps of both models. For the higher level feature, the channel attention module is capable of capture more accurate context information with relative uniform activation within the same semantic object.

Latent space analysis: As a generative model, our framework aims to estimate meaningful uncertainty map representing model awareness of its prediction. We analyse the uncertainty map of each

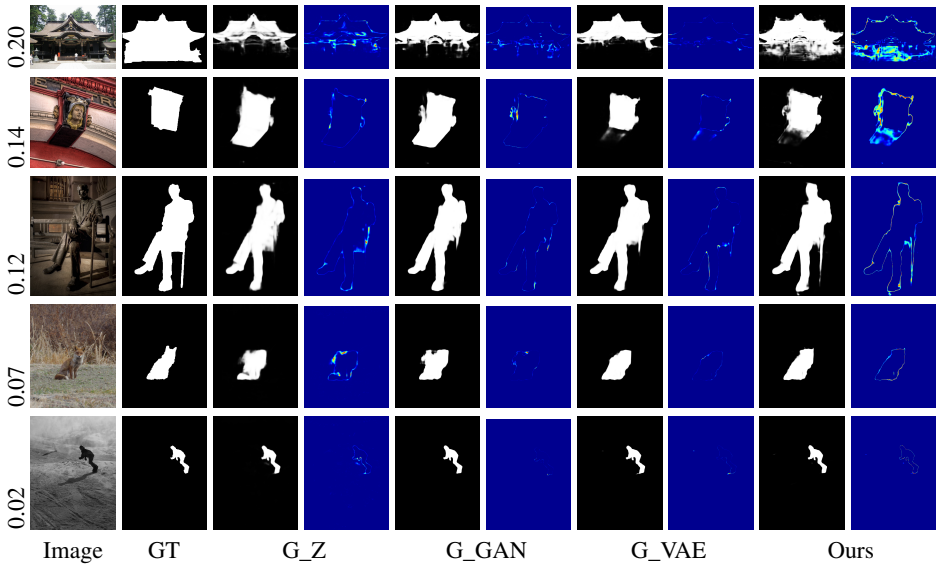

Figure 6: Model predictions and the uncertainty maps of the generative models and ours.

generative model in Table 4 w.r.t.complexity of the input image, and show results in Fig. 6. Our basic assumption is that a well calibrated model should generate confident predictions for easy samples with simple background, and less confident predictions for samples with complicated background.

To estimate complexity of the color image $\mathbf{I}$, we first over-segment it with SLIC [1] to generate $K = 200$ superpixels. Then, we compute similarity of each superpixel with all the other superpixels with handcrafted features from [91], which gives us a K dimensional contrast based vector, representing the overall image contrast. We define the saliency complexity of $\mathbf{I}$ as mean entropy of it's contrast vector. We find this contrast similarity based entropy measure is effective in evaluating the complexity of image for salient object detection, as images with similar contrast level are harder for salient object detection. In Fig. 6 of the first column, we show complexity of each image, which is consistent with human perception. Fig. 6 further shows that uncertainty maps from the existing generative models fail to explain model awareness of accuracy, *e.g.*the prediction of "G_GAN" is not accurate for the second image, while the uncertainty map cannot explain the less accurate prediction. Our uncertainty map on the contrary can effectively reflect accuracy of the prediction, which is also consistency with image complexity, *e.g.*easier images lead to higher confident predictions, and vice versa.

5 Conclusion

In this paper, we study the generative modeling of vision transformer for salient object detection. We start from defining a conditional probability distribution of saliency map given an image by a top-down latent variable generative framework, in which the non-linear mapping from image to saliency map is parameterized by a proposed vision transformer and the prior distribution of the low-dimensional latent space is represented by an energy-based model. Instead of using amortized inference and sampling strategies, we learn the model by MCMC-based maximum likelihood where Langevin sampling is used to evaluate the intractable posterior and prior distributions of the latent variables for calculating the gradients of the log-likelihood w.r.t.the model parameters. With the informative energy-based prior and the expressive top-down vision transformer, our model achieve both accurate predictions and meaningful uncertainty map that is consistent with human perception.

6 Broader Impact

The proposed generative vision transformer with energy-based latent space is effective in explaining data uncertainty, leading to more interpretable model, which is beneficial for the other higher level vision tasks, e.g., semantic segmentation, image caption and etc.

References

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurélien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels. *Technical report, EPFL*, 06 2010.

[2] Sanjeev Arora, Andrej Risteski, and Yi Zhang. Do gans learn the distribution? some theory and empirics. In *Int. Conf. Learn. Represent.*, 2018.

[3] Christian F. Baumgartner, Kerem Can Tezcan, Krishna Chaitanya, Andreas M. Hötker, Urs J. Muehlematter, Khoschy Schawkat, Anton S. Becker, Olivio Donati, and Ender Konukoglu. Phiseg: Capturing uncertainty in medical image segmentation. In *MICCAI*, pages 119–127, 2019.

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Eur. Conf. Comput. Vis.*, pages 213–229, 2020.

[5] Hao Chen and Youfu Li. Progressively complementarity-aware fusion network for RGB-D salient object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3051–3060, 2018.

[6] Hao Chen and Youfu Li. Three-stream attention-aware network for RGB-D salient object detection. *IEEE T. Image Process.*, 28(6):2825–2835, 2019.

[7] Hao Chen, Youfu Li, and Dan Su. Multi-modal fusion network with multi-scale multi-path and cross-modal interactions for RGB-D salient object detection. *Pattern Recognit.*, 86:376–385, 2019.

[8] Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, Philip HS Torr, and Shi-Min Hu. Global contrast based salient region detection. *IEEE T. Pattern Anal. Mach. Intell.*, 37(3):569–582, 2014.

[9] Yupeng Cheng, Huazhu Fu, Xingxing Wei, Jiangjian Xiao, and Xiaochun Cao. Depth enhanced saliency detection method. In *Proceedings of international conference on internet multimedia computing and service*, pages 23–27, 2014.

[10] Peng Chu, Jiang Wang, Quanzeng You, Haibin Ling, and Zicheng Liu. Spatial-temporal graph transformer for multiple object tracking. *arXiv preprint arXiv:2104.00194*, 2021.

[11] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. UP-DETR: Unsupervised pre-training for object detection with transformers. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 248–255, 2009.

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Int. Conf. Learn. Represent.*, 2021.

[14] Deng-Ping Fan, Ming-Ming Cheng, Yun Liu, Tao Li, and Ali Borji. Structure-measure: A new way to evaluate foreground maps. In *Int. Conf. Comput. Vis.*, pages 4548–4557, 2017.

[15] Deng-Ping Fan, Cheng Gong, Yang Cao, Bo Ren, Ming-Ming Cheng, and Ali Borji. Enhanced-alignment measure for binary foreground map evaluation. *arXiv preprint arXiv:1805.10421*, 2018.

[16] Deng-Ping Fan, Zheng Lin, Zhao Zhang, Menglong Zhu, and Ming-Ming Cheng. Rethinking RGB-D Salient Object Detection: Models, Datasets, and Large-Scale Benchmarks. *IEEE T. Neural Netw. Learn. Syst.*, 2020.

[17] Deng-Ping Fan, Yingjie Zhai, Ali Borji, Jufeng Yang, and Ling Shao. BBS-Net: RGB-D salient object detection with a bifurcated backbone strategy network. In *Eur. Conf. Comput. Vis.*, pages 275–292, 2020.

[18] Keren Fu, Deng-Ping Fan, Ge-Peng Ji, and Qijun Zhao. JL-DCF: Joint learning and densely-cooperative fusion framework for RGB-D salient object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3052–3062, 2020.

[19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Adv. Neural Inform. Process. Syst.*, pages 2672–2680, 2014.

[20] Rick Groenendijk, Sezer Karaoglu, Theo Gevers, and Thomas Mensink. On the benefit of adversarial training for monocular depth estimation. *Computer Vision and Image Understanding*, 190:102848, 2020.

[21] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Int. Conf. Mach. Learn.*, 2017.

[22] Junwei Han, Hao Chen, Nian Liu, Chenggang Yan, and Xuelong Li. CNNs-based RGB-D saliency detection via cross-view transfer and multiview fusion. *IEEE T. Cybern.*, 48(11):3171–3183, 2017.

[23] Tian Han, Yang Lu, Song-Chun Zhu, and Ying Nian Wu. Alternating back-propagation for generator network. In *AAAI Conf. Art. Intell.*, page 1976–1984, 2017.
[24] Junxian He, Daniel Spokoyny, Graham Neubig, and Taylor Berg-Kirkpatrick. Lagging inference networks and posterior collapse in variational autoencoders. In *Int. Conf. Learn. Represent.*, 2019.
[25] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.
[26] 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.
[27] Wei Ji, Jingjing Li, Miao Zhang, Yongri Piao, and Huchuan Lu. Accurate RGB-D salient object detection via collaborative learning. In *Eur. Conf. Comput. Vis.*, 2020.
[28] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and R. Hartley. Learning to estimate hidden motions with global motion aggregation. *arXiv preprint arXiv:2104.02409*, 2021.
[29] Ran Ju, Yang Liu, Tongwei Ren, Ling Ge, and Gangshan Wu. Depth-aware salient object detection using anisotropic center-surround difference. *Signal Processing: Image Communication*, 38:115 – 126, 2015.
[30] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
[31] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *Int. Conf. Learn. Represent.*, 2013.
[32] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
[33] Simon Kohl, Bernardino Romera-Paredes, Clemens Meyer, Jeffrey De Fauw, Joseph R. Ledsam, Klaus Maier-Hein, S. M. Ali Eslami, Danilo Jimenez Rezende, and Olaf Ronneberger. A probabilistic u-net for segmentation of ambiguous images. In *Adv. Neural Inform. Process. Syst.*, pages 6965–6975, 2018.
[34] Quang H. Le, Kamal Youcef-Toumi, Dzmitry Tsetserukou, and Ali Jahanian. GAN mask R-CNN: instance semantic segmentation benefits from generativeadversarial networks. *CoRR*, abs/2010.13757, 2020.
[35] Bo Li, Zhengxing Sun, and Yuqi Guo. Supervae: Superpixelwise variational autoencoder for salient object detection. In *AAAI Conf. Art. Intell.*, pages 8569–8576, 2019.
[36] Chongyi Li, Runmin Cong, Yongri Piao, Qianqian Xu, and Chen Change Loy. RGB-D salient object detection with cross-modality modulation and selection. In *Eur. Conf. Comput. Vis.*, pages 225–241, 2020.
[37] Gongyang Li, Zhi Liu, Linwei Ye, Yang Wang, and Haibin Ling. Cross-modal weighting network for RGB-D salient object detection. In *Eur. Conf. Comput. Vis.*, pages 665–681, 2020.
[38] Guanbin Li and Yizhou Yu. Visual saliency based on multiscale deep features. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5455–5463, 2015.
[39] Nianyi Li, Jinwei Ye, Yu Ji, Haibin Ling, and Jingyi Yu. Saliency detection on light field. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2806–2813, 2014.
[40] Yin Li, Xiaodi Hou, Christof Koch, James M Rehg, and Alan L Yuille. The secrets of salient object segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 280–287, 2014.
[41] Jiang-Jiang Liu, Qibin Hou, Ming-Ming Cheng, Jiashi Feng, and Jianmin Jiang. A simple pooling-based design for real-time salient object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.
[42] Jun S Liu. *Monte Carlo strategies in scientific computing*. Springer Science & Business Media, 2008.
[43] Nian Liu, Junwei Han, and Ming-Hsuan Yang. Picanet: Learning pixel-wise contextual attention for saliency detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3089–3098, 2018.
[44] Nian Liu, Ni Zhang, and Junwei Han. Learning selective self-mutual attention for RGB-D saliency detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 13756–13765, 2020.
[45] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.
[46] Zhiming Luo, Akshaya Mishra, Andrew Achkar, Justin Eichel, Shaozi Li, and Pierre-Marc Jodoin. Non-local deep features for salient object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, July 2017.
[47] Weian Mao, Yongtao Ge, Chunhua Shen, Zhi Tian, Xinlong Wang, and Zhibin Wang. TFPose: Direct human pose estimation with transformers. *arXiv preprint arXiv:2103.15320*, 2021.
[48] Vida Movahedi and James H Elder. Design and perceptual validation of performance measures for salient object segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, pages

49–56, 2010.
[49] Radford M Neal et al. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2, 2011.
[50] Erik Nijkamp, Mitch Hill, Song-Chun Zhu, and Ying Nian Wu. Learning non-convergent non-persistent short-run mcmc toward energy-based model. *arXiv preprint arXiv:1904.09770*, 2019.
[51] Erik Nijkamp, Bo Pang, Tian Han, Linqi Zhou, Song-Chun Zhu, and Ying Nian Wu. Learning multi-layer latent variable model via variational optimization of short run mcmc for approximate inference. In *European Conference on Computer Vision*, pages 361–378. Springer, 2020.
[52] Yuzhen Niu, Yujie Geng, Xueqing Li, and Feng Liu. Leveraging stereopsis for saliency analysis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 454–461, 2012.
[53] Bo Pang, Tian Han, Erik Nijkamp, Song-Chun Zhu, and Ying Nian Wu. Learning latent space energy-based prior model. In *Adv. Neural Inform. Process. Syst.*, pages 21994–22008, 2020.
[54] Youwei Pang, Lihe Zhang, Xiaoqi Zhao, and Huchuan Lu. Hierarchical dynamic filtering network for RGB-D salient object detection. In *Eur. Conf. Comput. Vis.*, 2020.
[55] Houwen Peng, Bing Li, Weihua Xiong, Weiming Hu, and Rongrong Ji. RGBD salient object detection: A benchmark and algorithms. In *Eur. Conf. Comput. Vis.*, pages 92–109, 2014.
[56] Yongri Piao, Wei Ji, Jingjing Li, Miao Zhang, and Huchuan Lu. Depth-induced multi-scale recurrent attention network for saliency detection. In *Int. Conf. Comput. Vis.*, pages 7254–7263, 2019.
[57] Yongri Piao, Zhengkun Rong, Miao Zhang, Weisong Ren, and Huchuan Lu. A2dele: Adaptive and attentive depth distiller for efficient RGB-D salient object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9060–9069, 2020.
[58] Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jagersand. Basnet: Boundary-aware salient object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7479–7489, 2019.
[59] Liangqiong Qu, Shengfeng He, Jiawei Zhang, Jiandong Tian, Yandong Tang, and Qingxiong Yang. RGBD salient object detection via deep fusion. *IEEE T. Image Process.*, 26(5):2274–2285, 2017.
[60] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *arXiv preprint arXiv:2103.13413*, 2021.
[61] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.
[62] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Adv. Neural Inform. Process. Syst.*, pages 3483–3491, 2015.
[63] Nasim Souly, Concetto Spampinato, and Mubarak Shah. Semi supervised semantic segmentation using generative adversarial network. In *Int. Conf. Comput. Vis.*, pages 5689–5697, 2017.
[64] Lucas Stoffl, Maxime Vidal, and Alexander Mathis. End-to-End trainable Multi-Instance pose estimation with transformers. *arXiv preprint arXiv:2103.12115*, 2021.
[65] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Adv. Neural Inform. Process. Syst.*, 2017.
[66] Bo Wang, Quan Chen, Min Zhou, Zhiqiang Zhang, Xiaogang Jin, and Kun Gai. Progressive feature polishing network for salient object detection. In *AAAI Conf. Art. Intell.*, pages 12128–12135, 2020.
[67] Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. Learning to detect salient objects with image-level supervision. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 136–145, 2017.
[68] Ningning Wang and Xiaojin Gong. Adaptive fusion for RGB-D salient object detection. *IEEE Access*, 7:55277–55284, 2019.
[69] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *arXiv preprint arXiv:2102.12122*, 2021.
[70] Jun Wei, Shuhui Wang, and Qingming Huang. F^3net: Fusion, feedback and focus for salient object detection. In *AAAI Conf. Art. Intell.*, pages 12321–12328, 2020.
[71] Jun Wei, Shuhui Wang, Zhe Wu, Chi Su, Qingming Huang, and Qi Tian. Label decoupling framework for salient object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 13025–13034, 2020.
[72] Zhe Wu, Li Su, and Qingming Huang. Cascaded partial decoder for fast and accurate salient object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2019.

[73] Zhe Wu, Li Su, and Qingming Huang. Stacked cross refinement network for edge-aware salient object detection. In *Int. Conf. Comput. Vis.*, pages 7264–7273, 2019.
[74] Jianwen Xie, Ruiqi Gao, Zilong Zheng, Song-Chun Zhu, and Ying Nian Wu. Learning dynamic generator model by alternating back-propagation through time. In *AAAI Conf. Art. Intell.*, pages 5498–5507, 2019.
[75] Jianwen Xie, Yang Lu, Ruiqi Gao, Song-Chun Zhu, and Ying Nian Wu. Cooperative training of descriptor and generator networks. *IEEE transactions on pattern analysis and machine intelligence*, 42(1):27–45, 2018.
[76] Jianwen Xie, Yang Lu, Song-Chun Zhu, and Yingnian Wu. A theory of generative convnet. In *Int. Conf. Mach. Learn.*, pages 2635–2644, 2016.
[77] Yihong Xu, Yutong Ban, Guillaume Delorme, Chuang Gan, Daniela Rus, and Xavier Alameda-Pineda. TransCenter: Transformers with dense queries for multiple-object tracking. *arXiv preprint arXiv:2103.15145*, 2021.
[78] Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. Learning spatio-temporal transformer for visual tracking. *arXiv preprint arXiv:2103.17154*, 2021.
[79] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1155–1162, 2013.
[80] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3166–3173, 2013.
[81] Gongjie Zhang, Zhipeng Luo, Kaiwen Cui, and Shijian Lu. Meta-DETR: Few-shot object detection via unified image-level meta-learning. *arXiv preprint arXiv:2103.11731*, 2021.
[82] Jing Zhang, Deng-Ping Fan*, Yuchao Dai, Saeed Anwar, Fatemeh Sadat Saleh, Sadegh Aliakbarian, and Nick Barnes. Uncertainty inspired rgb-d saliency detection. *IEEE T. Pattern Anal. Mach. Intell.*, 2021.
[83] Jing Zhang, Deng-Ping Fan, Yuchao Dai, Saeed Anwar, Fatemeh Sadat Saleh, Tong Zhang, and Nick Barnes. Uc-net: Uncertainty inspired rgb-d saliency detection via conditional variational autoencoders. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.
[84] Miao Zhang, Weisong Ren, Yongri Piao, Zhengkun Rong, and Huchuan Lu. Select, supplement and focus for RGB-D saliency detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3472–3481, 2020.
[85] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Eur. Conf. Comput. Vis.*, pages 286–301, 2018.
[86] Zhao Zhang, Zheng Lin, Jun Xu, Wen-Da Jin, Shao-Ping Lu, and Deng-Ping Fan. Bilateral attention network for RGB-D salient object detection. *IEEE T. Image Process.*, 30:1949–1961, 2021.
[87] Jia-Xing Zhao, Yang Cao, Deng-Ping Fan, Ming-Ming Cheng, Xuan-Yi Li, and Le Zhang. Contrast prior and fluid pyramid integration for RGBD salient object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3927–3936, 2019.
[88] Jia-Xing Zhao, Jiang-Jiang Liu, Deng-Ping Fan, Yang Cao, Jufeng Yang, and Ming-Ming Cheng. Egnet:edge guidance network for salient object detection. In *Int. Conf. Comput. Vis.*, Oct 2019.
[89] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.
[90] Huajun Zhou, Xiaohua Xie, Jian-Huang Lai, Zixuan Chen, and Lingxiao Yang. Interactive two-stream decoder for accurate and fast saliency detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9141–9150, 2020.
[91] Wangjiang Zhu, Shuang Liang, Yichen Wei, and Jian Sun. Saliency optimization from robust background detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2814–2821, 2014.
[92] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *Int. Conf. Learn. Represent.*, 2021.

Checklist

1. For all authors...

   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] We have claim our contributions and scope in Section 1

(b) Did you describe the limitations of your work? [Yes] We describe the limitation of our model that is more time-consuming than generative models with amortized inference.

(c) Did you discuss any potential negative societal impacts of your work? [Yes] We have provided the explanation in the "Broader Impact" section and we believe that this work will not adversely affect the society.

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] We have read through the ethics review guidelines and confirm that our paper conforms to them.

2. If you are including theoretical results...

(a) Did you state the full set of assumptions of all theoretical results? [N/A]

(b) Did you include complete proofs of all theoretical results? [N/A]

3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We will include the trained model, instructions and experimental results in the supplemental materials.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We specify all the training details in Section 4.1 and the "Hyper-parameter analysis" section.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Due to space limitation, we will report error bars in the supplemental materials.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We specify all the training details in Section 4.1.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] We use open source datasets for training and testing our SOD and RGB-D SOD tasks. The details of datasets can be seen in Section 4.1.

(b) Did you mention the license of the assets? [N/A] We reference the PyTorch open source code of Swin Transformer[45] and its pre-train model, which is MIT License.

(c) Did you include any new assets either in the supplemental material or as a URL? [No] We don't have any new assets either in the supplemental material or as a URL.

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] The data used in our paper does not involve this aspect.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] The data used in our paper does not involve this aspect.

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] We don't use crowdsourcing or conducted research with human subjects.

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] We don't use crowdsourcing or conducted research with human subjects.

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] We don't use crowdsourcing or conducted research with human subjects.