Unsupervised Foreground Extraction via Deep Region Competition

Anonymous Author(s)
Affiliation
Address
email

Abstract

We present Deep Region Competition (DRC), an algorithm designed to extract foreground objects from images in a fully unsupervised manner. Foreground extraction can be viewed as a special case of generic image segmentation that focuses on identifying and disentangling objects from the background. In this work, we rethink the foreground extraction by reconciling energy-based prior with generative image modeling in the form of Mixture of Experts (MoE), where we further introduce the learned pixel re-assignment as the essential inductive bias to capture the regularities of background regions. With this modeling, the foreground-background partition can be naturally found through Expectation-Maximization (EM). We show that the proposed method effectively exploits the interaction between the mixture components during the partitioning process, which closely connects to region competition [1], a seminal approach for generic image segmentation. Experiments demonstrate that DRC exhibits more competitive performances on complex real-world data and challenging multi-object scenes compared with prior methods. Moreover, we show empirically that DRC can potentially generalize to novel foreground objects even from categories unseen during training.

1 Introduction

Foreground extraction, being a special case of generic image segmentation, aims for a binary partition of the given image with specific semantic meaning, *i.e.*, a foreground that typically contains identifiable objects and the possibly less structural remaining regions as the background. There is a rich literature on explicitly modeling and representing a given image as foreground and background (or more general visual regions), such that a generic inference algorithm can produce plausible segmentations ideally for any images without or with little supervision [1–8]. However, such methods essentially rely on low-level visual features (*e.g.*, edges, color, and texture), and some further require human intervention at initialization [4, 5], which largely limits their practical performance on modern datasets with complex natural images and rich semantic meanings [9, 10]. These datasets typically come with fine-grained semantic annotations, exploited by supervised methods that learn representation and inference algorithm as one monolithic network [11–16]. Despite the success of densely supervised learning, the unsupervised counterpart is still favored due to its resemblance to how humans perceive the world [17, 18].

Attempting to combine unsupervised learning or weakly supervised learning with modern neural networks, three lines of work surge recently for foreground extraction: (1) deep networks as feature extractors for canonical segmentation algorithms; (2) GAN-based foreground-background disentanglement; (3) compositional latent variable models with slot-based object modeling. Albeit the amazing designs and vast successes of these recently emerged methods, the question of foreground extraction without supervision remains open. Works from the first line train designated deep feature

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

extractors for canonical segmentation algorithms or metric networks as learned partitioning criteria [19–22]. These methods, such as W-Net [19], define the properties for foreground objects using learned features or criteria and are thus generally bottle-necked by the selected postprocessing segmentation algorithm [23, 24]. GAN-based models [25–30] capture the foreground objectness with oversimplified assumptions, or require additional supervision to achieve foreground-background disentanglement. For example, the segmentation model in ReDO [27] is trained by redrawing detected objects, which could potentially limit its application to datasets where object shapes vary significantly. OneGAN [30] and its predecessors [28, 29], though producing impressive results on foreground extraction, require a set of background images without foreground objects as additional inputs for extra supervision. Lastly, compositional latent variable models [31–39] include the background as a "virtual object," and induce the independence of object representations using an identical generator for all object slots. Although these methods exhibit strong performance on synthetic multi-object datasets with simple backgrounds and foreground shapes, they may fail on complex real-world data or even synthetic datasets with more challenging backgrounds [32, 33]. Moreover, few unsupervised learning methods have provided *explicit identification of foreground objects and background regions*. While being capable of generating valid segmentation masks, most of these methods do not specify which output corresponds to the foreground objects. These deficiencies necessitate rethinking the problem of unsupervised foreground extraction. We propose to confront the challenges in formulating (1) a generic inductive bias for modeling foreground and background regions that can be baked into neural generators; and (2) an effective inference algorithm based on a principled criterion for foreground-background partition.

Inspired by Region Competition [1], a seminal approach for generic image segmentation that combines optimization-based inference [40–42], probabilistic visual modeling [43, 44] and a Bayesian criterion [45], we propose to solve the foreground extraction problem by reconciling energy-based prior [46] with generative image modeling in the form of Mixture of Experts (MoE) [47, 48] in this work. As an attempt to provide generic descriptions for background regions, we further introduce the learned pixel re-assignment as the essential inductive bias to capture the background regularities. Fueled by our modeling, we propose to find the foreground-background partition through Expectation-Maximization (EM). The devised algorithm effectively exploits the interaction between the mixture components during the partitioning process, which resembles the intuition described in Region Competition [1]. We therefore coin our method Deep Region Competition (DRC). We summarize our **contributions** as follows:

1. We provide probabilistic foreground-background modeling by reconciling energy-based prior with generative image modeling in the form of MoE. With this modeling, the foreground-background partition can be naturally produced through EM. We further introduce an inductive bias, *pixel re-assignment*, to facilitate foreground-background disentanglement.
2. In experiments, we demonstrate that DRC exhibits more competitive performances on complex real-world data and challenging multi-object scenes compared with prior methods. Further, we empirically show that using learned pixel re-assignment as the inductive bias helps to provide explicit identification for foreground and background regions.
3. We find that DRC can potentially generalize to novel foreground objects even from categories unseen during training, which may provide some inspiration for the study of out-of-distribution (OOD) generalization in more general unsupervised disentanglement.

2 Preliminaries

Latent-space Energy-based Model Adopting the language of EM algorithm, we assume that for the observed sample $\mathbf{x} \in \mathbb{R}^D$, there exists $\mathbf{z} \in \mathbb{R}^d$ as its latent variables. The complete-data distribution is

$$p_\theta(\mathbf{z}, \mathbf{x}) = p_\alpha(\mathbf{z}) p_\beta(\mathbf{x}|\mathbf{z}), \tag{1}$$

where $p_\alpha(\mathbf{z})$ is the prior model with parameters α, $p_\beta(\mathbf{x}|\mathbf{z})$ is the top-down generative model with parameters β, and $\theta = (\alpha, \beta)$.

The prior model $p_\alpha(\mathbf{z})$ can be formulated as an energy-based model, which we refer to as the Latent-space Energy-Based Model (LEBM) [46] throughout the paper:

$$p_\alpha(\mathbf{z}) = \frac{1}{Z_\alpha} \exp\left(f_\alpha(\mathbf{z})\right) p_0(\mathbf{z}), \tag{2}$$

where $p_0(\mathbf{z})$ is a reference distribution, assumed to be isotropic Gaussian prior commonly used for the generative model, $f_\alpha(\mathbf{z})$ can be parameterized by a neural network, and Z_α is the partition function. The prior model in Eq. (2) can be interpreted as an energy-based correction or exponential tilting of the original prior distribution p_0.

The LEBM can be learned by Maximum Likelihood Estimation (MLE). Given a training sample $\mathbf{x}$, the learning gradient for α is derived as shown by Pang et al. [46],

$$\delta_\alpha(\mathbf{x}) = \mathbf{E}_{p_\theta(\mathbf{z}|\mathbf{x})}\left[\nabla_\alpha f_\alpha(\mathbf{z})\right] - \mathbf{E}_{p_\alpha(\mathbf{z})}\left[\nabla_\alpha f_\alpha(\mathbf{z})\right]. \tag{3}$$

In practice, the above expectations can be approximated by Monte-Carlo average, which requires sampling from $p_\theta(\mathbf{z}|\mathbf{x})$ and $p_\alpha(\mathbf{z})$. This step can be done with stochastic gradient-based methods, such as Langevin dynamics [49] or Hamiltonian Monte Carlo [50].

LEBM with Symbol-Vector Duality An extension to the above LEBM is to further couple the vector representation $\mathbf{z}$ with a symbolic representation $\mathbf{y}$ [51]. Formally, $\mathbf{y}$ is a K-dimensional one-hot vector, where K is the number of possible $\mathbf{z}$ categories. Such symbol-vector duality can provide extra entries for auxiliary supervision; we will detail in Section 3.3.

The LEBM with symbol-vector duality is formulated as

$$p_\alpha(\mathbf{y}, \mathbf{z}) = \frac{1}{Z_\alpha} \exp\left(\langle \mathbf{y}, f_\alpha(\mathbf{z})\rangle\right) p_0(\mathbf{z}), \tag{4}$$

where $f_\alpha(\mathbf{z}) \in \mathbb{R}^K$ is the joint energy term. To directly infer $\mathbf{z}$ from $\mathbf{x}$, we can marginalize $\mathbf{y}$ out to obtain an energy-based prior as in Eq. (2): $p_\alpha(\mathbf{z}) = \frac{1}{Z_\alpha}\exp(F_\alpha(\mathbf{z}))p_0(\mathbf{z})$, where $F_\alpha(\mathbf{z}) = \log \sum_{\mathbf{y}} \exp\left(\langle \mathbf{y}, f_\alpha(\mathbf{z})\rangle\right)$ is the marginal energy term. After inferring $\mathbf{z}$ from $\mathbf{x}$, *i.e.*, $p_\alpha(\mathbf{z})$ is fixed, we have $p_\alpha(\mathbf{y}|\mathbf{z}) \propto \exp(\langle \mathbf{y}, f_\alpha(\mathbf{z})\rangle$ that naturally forms a softmax classifier of $\mathbf{z}$, and $f_\alpha(\mathbf{z})$ provides the K logit scores of the K categories.

3 Methodology

Canonical segmentation approaches decompose a given image $\mathbf{I}$ into a set of n non-overlapping regions. Foreground extraction, however, focuses on extracting the foreground region from the image, *i.e.*, performing a binary partition for the given image $\mathbf{I}$. Without explicit supervision, we propose to use learned pixel re-assignment as a generic inductive bias for background modeling, upon which we derive an EM-like partitioning algorithm. Compared with methods based primarily on handcrafted low-level features, our learning-based modeling approach can handle more complex natural images. Moreover, our algorithm demonstrates explicit identification of foreground and background regions, a capability that few unsupervised learning methods have shown in the literature.

This section is organized as follows: we detail the image modeling in Section 3.1 and introduce our algorithm in Section 3.2. Implementation details are provided in Section 3.3.

3.1 Generative Image Modeling

Mixture of Experts for Image Generation Inspired by the regional homogenity assumption proposed by Zhu and Yuille [1], we use separate priors and generative models for foreground and background regions, indexed as α_k and β_k, $k = 1, 2$ respectively (see Fig. 1). This design leads to the form of MoE [47, 48] for image modeling, as shown below.

Let us start by considering only the i-th pixel of the observed image $\mathbf{x}$, denoted as $\mathbf{x}_i$. We use a binary one-hot random variable $\mathbf{w}_i$ to indicate whether the i-th pixel belongs to the foreground region. Formally, we have $\mathbf{w}_i = [w_{i1}, w_{i2}]$, $w_{ik} \in \{0, 1\}$ and $\sum_{k=1}^{2} w_{ik} = 1$. Let $w_{i1} = 1$ indicate that the i-th pixel $\mathbf{x}_i$ belongs to the foreground, and $w_{i2} = 1$ indicate that the i-th pixel $\mathbf{x}_i$ belongs to the background. We assume that the distribution of $\mathbf{w}_i$ is prior-dependent. Specifically, the mixture parameter $\pi_{ik},\ k = 1, 2$ is defined as the output of a parameterized gating function $\pi_{ik} = p_\beta(w_{ik} = 1|\mathbf{z}) = \text{Softmax}(l_{ik})$, where $l_{ik} = h_{\beta_k}(\mathbf{z}_k), k = 1, 2$ are the logit scores given by the corresponding generative models, and $\beta = \{\beta_1, \beta_2\}, \mathbf{z} = \{\mathbf{z}_1, \mathbf{z}_2\}$. Taken together, the joint distribution of $\mathbf{w}_i$ is

$$p_\beta(\mathbf{w}_i|\mathbf{z}) = \prod_{k=1}^{2} \pi_{ik}^{w_{ik}}. \tag{5}$$

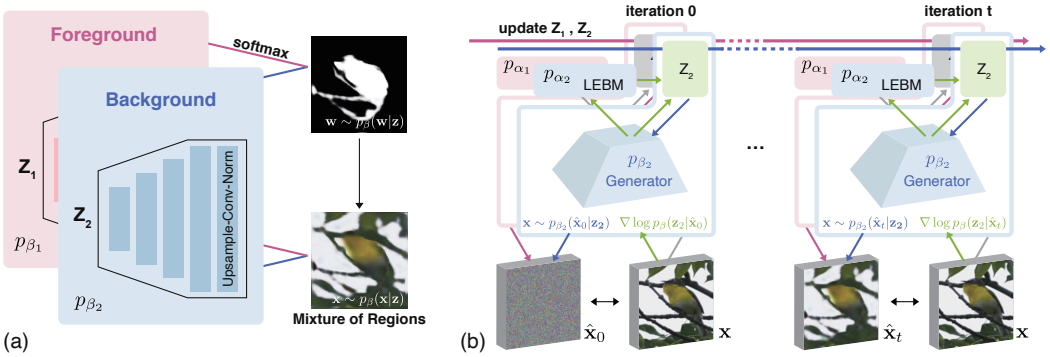

Figure 1: **Overview of DRC.** (a) The model generates foreground and background regions using sampled latent variables $\mathbf{z} = \{\mathbf{z}_1, \mathbf{z}_2\}$. $p_{\beta_k},\ k = 1, 2$ represents the generator for each region. (b) DRC samples the latent variables $\mathbf{z}$ in an iterative manner. Let $\mathbf{x}$ denote the observed image; we use $\hat{\mathbf{x}}_t,\ t = 0, 1, \ldots$ to represent the image generated by $p_\beta(\mathbf{x}|\mathbf{z})$ at the t-th sampling step. DRC has a two-step workflow for learning unsupervised foreground extractors that resembles the E- and M-step in the classic EM algorithm. In the E-step, it employs gradient-based MCMC sampling to infer the latent variables $\mathbf{z}$ as shown in (b). Of note, only the latent variables $\mathbf{z}$ are updated in this step. In the M-step, the sampled latent variables $\mathbf{z}$ are fed into the model for image generation (as shown in (a)), where the generators are updated to minimize the reconstruction error.

Generating $\mathbf{x}_i$ based on the distribution of $\mathbf{w}_i$ involves two steps: (1) sample $\mathbf{w}_i$ from the distribution $p_\beta(\mathbf{w}_i|\mathbf{z})$, and (2) choose either the foreground model (*i.e.*, $p_{\beta_1}(\mathbf{x}_i|w_{i1} = 1, \mathbf{z}_1)$) or the background model (*i.e.*, $p_{\beta_2}(\mathbf{x}_i|w_{i2} = 1, \mathbf{z}_2)$) to generate $\mathbf{x}_i$ based on the value of sampled $\mathbf{w}_i$. This distribution of $\mathbf{x}_i$ is therefore a MoE,

$$p_\beta(\mathbf{x}_i|\mathbf{z}) = \sum_{k=1}^{2} p_\beta(w_{ik} = 1|\mathbf{z}) p_{\beta_k}(\mathbf{x}_i|w_{ik} = 1, \mathbf{z}_k) = \sum_{k=1}^{2} \pi_{ik} p_{\beta_k}(\mathbf{x}_i|\mathbf{z}_k), \tag{6}$$

wherein the posterior responsibility of w_{ik} is

$$\gamma_{ik} = p(w_{ik} = 1|\mathbf{x}_i, \mathbf{z}) = \frac{\pi_{ik} p_{\beta_k}(\mathbf{x}_i|\mathbf{z}_k)}{\sum_{m=1}^{2} \pi_{im} p_{\beta_m}(\mathbf{x}_i|\mathbf{z}_m)},\ k = 1, 2. \tag{7}$$

Using a fully-factorized joint distribution of $\mathbf{x}$, we have $p_\beta(\mathbf{x}|\mathbf{z}) = \prod_{i=1}^{D} \sum_{k=1}^{2} \pi_{ik} p_{\beta_k}(\mathbf{x}_i|\mathbf{z}_k)$ as the generative modeling of $\mathbf{x} \in \mathbb{R}^D$. In particular, we assume that the generative model $p_{\beta_k}(\mathbf{x}_i|\mathbf{z}_k),\ k = 1, 2$ follows a Gaussian distribution.

Learning Pixel Re-assignment for Background Modeling We use pixel re-assignment in the background generative model as the key inductive bias for modeling the background region. This is partially inspired by the concepts of "texture" and "texton" by Julez [44, 52], where the textural part of an image may contain fewer structural elements in preattentive vision, which coincides with our intuitive observation of the background regions.

We use a separate pair of energy-based prior model α_p and generative model β_p to learn the re-assignment. For simplicity, we absorb α_p and β_p in the models for background modeling, *i.e.*, α_2 and β_2 respectively. In practice, the re-assignment follows the output of β_p, a shuffling grid with the same size of the image $\mathbf{x}$. Its values indicate the re-assigned pixel coordinates; see Fig. 2.

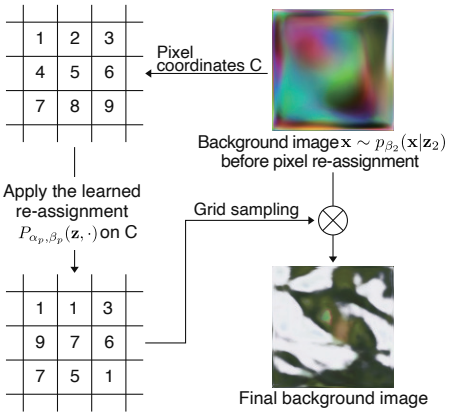

Figure 2: **Pixel re-assignment.** The output of β_p can be viewed as a learned re-assignment of the original background pixels that follows the mapped grid $P_{\alpha_p,\beta_p}(\mathbf{z}, C)$. Note that the re-assignment function $P_{\alpha_p,\beta_p}(\mathbf{z}, \cdot)$ might not be injective. The final background image is generated via grid sampling.

We find that shuffling the background pixels using the learned re-assignment facilitates the model to capture the regularities of the background regions. Specifically, the proposed model with this key inductive bias learns to constantly give the correct mask assignment, whereas most previous fully unsupervised methods do not provide explicit identification of the foreground and background regions; see discussion in Section 4.1 for more details.

3.2 Deep Region Competition: from Generative Modeling to Foreground Extraction

The complete-data distribution from the image modeling is

$$\begin{aligned} p_\theta(\mathbf{x}, \mathbf{z}, \mathbf{w}) &= p_\beta(\mathbf{x}|\mathbf{w}, \mathbf{z}) p_\beta(\mathbf{w}|\mathbf{z}) p_\alpha(\mathbf{z}) \\ &= \left(\prod_{i=1}^{D} \prod_{k=1}^{2} p_{\beta_k}(\mathbf{x}_i|\mathbf{z}_k)^{w_{ik}} \right) \left(\prod_{i=1}^{D} \prod_{k=1}^{2} \pi_{ik}^{w_{ik}} \right) p_\alpha(\mathbf{z}) \\ &= p_\alpha(\mathbf{z}) \prod_{i=1}^{D} \prod_{k=1}^{2} \left(\pi_{ik} p_{\beta_k}(\mathbf{x}_i|\mathbf{z}_k) \right)^{w_{ik}}, \end{aligned} \tag{8}$$

where $p_\alpha(\mathbf{z}) = p_{\alpha_1}(\mathbf{z}_1) p_{\alpha_2}(\mathbf{z}_2)$ is the prior model given by LEBMs. $\alpha = \{\alpha_1, \alpha_2\}$, and $\theta = \{\alpha, \beta\}$. $\mathbf{w}$ is the vector of $(\mathbf{w}_i), i = 1, ...D$, whose joint distribution is assumed to be fully-factorized.

Next, we derive the complete-data log-likelihood as our learning objective:

$$\mathcal{L}(\theta) = \log p_\theta(\mathbf{x}, \mathbf{z}, \mathbf{w}) = \log p_\alpha(\mathbf{z}) + \sum_{i=1}^{D} \sum_{k=1}^{2} w_{ik} \left(\log \pi_{ik} + \log p_{\beta_k}(\mathbf{x}_i|\mathbf{z}_k) \right). \tag{9}$$

Note that $\mathbf{w}$ and $\mathbf{z}$ are unobserved variables in the modeling, which makes it impossible to learn the model directly through MLE. To calculate the gradients of θ, we instead optimize $\mathbf{E}_{\mathbf{z}\sim p(\mathbf{z}|\mathbf{x}), \mathbf{w}\sim p(\mathbf{w}|\mathbf{x},\mathbf{z})}[\mathcal{L}(\theta)]$ based on the fact that underlies the EM algorithm: $\nabla_\theta \log p_\theta(\mathbf{x}) = \int_{\mathbf{z}} p_\theta(\mathbf{z}|\mathbf{x}) d\mathbf{z} \int_{\mathbf{w}} p_\theta(\mathbf{w}|\mathbf{z}, \mathbf{x}) \nabla_\theta \log p_\theta(\mathbf{x}, \mathbf{z}, \mathbf{w}) d\mathbf{w} = \mathbf{E}_{\mathbf{z}\sim p_\theta(\mathbf{z}|\mathbf{x}), \mathbf{w}\sim p_\theta(\mathbf{w}|\mathbf{x},\mathbf{z})}[\nabla_\theta \log p_\theta(\mathbf{x}, \mathbf{z}, \mathbf{w})]$.

Therefore, the derived surrogate learning objective becomes

$$\max_\theta \ \mathbf{E}_{\mathbf{z}\sim p_\theta(\mathbf{z}|\mathbf{x})} \left[\mathcal{J}(\theta) \right], \text{ s.t. } \forall i, \sum_{k=1}^{2} \pi_{ik} = 1, \tag{10}$$

$$\mathcal{J}(\theta) = \underbrace{\log p_\alpha(\mathbf{z})}_{\text{objective for LEBM}} + \underbrace{\sum_{i=1}^{D} \sum_{k=1}^{2} \gamma_{ik} \log \pi_{ik}}_{\text{foreground-background partitioning}} + \underbrace{\sum_{i=1}^{D} \sum_{k=1}^{2} \gamma_{ik} \log p_{\theta_k}(\mathbf{x}_i|\mathbf{z}_k)}_{\text{objective for image generation}}, \tag{11}$$

where $\mathcal{J}(\theta) = \mathbf{E}_{\mathbf{w}\sim p_\theta(\mathbf{w}|\mathbf{x},\mathbf{z})} \left[\mathcal{L}(\theta) \right]$ is the conditional expectation of $\mathbf{w}$, which can be calculated in closed form; see supplementary material for additional details.

Eq. (10) has an intuitive interpretation. We can decompose the learning objective into three parts as in Eq. (11). In particular, the second term $\sum_{i=1}^{D} \sum_{k=1}^{2} \gamma_{ik} \log \pi_{ik}$ has a similar form to the cross-entropy loss commonly used for supervised segmentation task, where the posterior responsibility γ_{ik} serves as the target distribution. It is as if the foreground and background generative models compete with each other to fit the distribution of each pixel $\mathbf{x}_i$. If the pixel value at $\mathbf{x}_i$ fits better to the distribution of foreground, $p_{\beta_1}(\mathbf{x}_i|\mathbf{z}_1)$, than to that of background, $p_{\beta_2}(\mathbf{x}_i|\mathbf{z}_2)$, the model tends to assign that pixel to the foreground region (see Eq. (7)), and vice versa. This mechanism is similar to the process derived in Zhu and Yuille [1], which is the reason why we coin our method Deep Region Competition (DRC).

Prior to our proposal, several works [1, 32, 33] also employ mixture models and competition among the components to perform unsupervised foreground or image segmentation. The original Region Competition [1] combines several families of image modeling with Bayesian inference, but is limited by the expressiveness of the pre-specified probability distributions. More recent works, including IODINE [32] and Slot-attention [33], learn amortized inference networks for latent variables, and induce the independence of foreground and background representations using an identical generator. Our method combines the best of the two worlds, reconciling the expressiveness of learned generators with the regularity of generic texture modeling under the framework of LEBM.

To optimize the learning objective in Eq. (10), we approximate the expectation by sampling from the prior $p_\alpha(\mathbf{z})$ and posterior model $p_\theta(\mathbf{z}|\mathbf{x}) \propto p_\alpha(\mathbf{z}) p_\beta(\mathbf{x}|\mathbf{z})$, and calculating the Monte Carlo average. We use Langevin dynamics [49] to draw persistent MCMC samples, which iterates

$$\mathbf{z}_{t+1} = \mathbf{z}_t + s \nabla_{\mathbf{z}} \log Q(\mathbf{z}_t) + \sqrt{2s} \epsilon_t, \tag{12}$$

Algorithm 1: Learning models of DRC via EM.

Input: Learning iterations T, initial parameters for LEBMs $\alpha^{(0)} = \{\alpha_1^{(0)}, \alpha_2^{(0)}\}$ and generators $\beta^{(0)} = \{\beta_1^{(0)}, \beta_2^{(0)}\}$, $\theta^{(0)} = \{\alpha^{(0)}, \beta^{(0)}\}$, learning rate η_α for LEBMs, η_β for foreground and background generators, observed examples $\{\mathbf{x}^{(i)}\}_{i=1}^N$, batch size M, and initial latent variables $\{\mathbf{z}_-^{(i)} = \{\mathbf{z}_{1-}^{(i)}, \mathbf{z}_{2-}^{(i)}\} \sim p_0(\mathbf{z})\}_{i=1}^N$ and $\{\mathbf{z}_+^{(i)} = \{\mathbf{z}_{1+}^{(i)}, \mathbf{z}_{2+}^{(i)}\} \sim p_0(\mathbf{z})\}_{i=1}^N$.

Output: $\theta^{(T)} = \{\alpha_1^{(T)}, \beta_1^{(T)}, \alpha_2^{(T)}, \beta_2^{(T)}\}$.

for $t = 0 : T-1$ **do**

  Sample a minibatch of data $\{\mathbf{x}^{(i)}\}_{i=1}^M$;

  Prior sampling for learning LEBMs: For each $\mathbf{x}^{(i)}$, update $\mathbf{z}_-^{(i)}$ using Eq. (12), with target distribution $\pi(\mathbf{z}) = p_{\alpha^{(t)}}(\mathbf{z})$;

  Posterior sampling for foreground and background generation: For each $\mathbf{x}^{(i)}$, update $\mathbf{z}_+^{(i)}$ using Eq. (12), with target distribution $\pi(\mathbf{z}) = p_{\theta^{(t)}}(\mathbf{z}|\mathbf{x})$;

  Update LEBMs: $\alpha^{(t+1)} = \alpha^{(t)} + \eta_\alpha \frac{1}{m} \sum_{i=1}^m [\nabla_\alpha f_{\alpha^{(t)}}(\mathbf{z}_+^{(i)}) - \nabla_\alpha f_{\alpha^{(t)}}(\mathbf{z}_-^{(i)})]$;

  Update foreground and background generators:
  $\beta^{(t+1)} = \beta^{(t)} + \eta_\beta \frac{1}{m} \sum_{i=1}^m \nabla_\beta \log p_{\beta^{(t)}}(\mathbf{x}^{(i)}|\mathbf{z}_+^{(i)})$;

where t indexes the time step of the Langevin dynamics, s represents the step size, and ϵ_t is the Gaussian white noise. $Q(\mathbf{z})$ denotes the target distribution, being either $p_\alpha(\mathbf{z})$ or $p_\theta(\mathbf{z}|\mathbf{x})$. $\nabla_\mathbf{z} \log Q(\mathbf{z}_t)$ can be efficiently computed via automatic differentiation in modern deep learning libraries [53]. We summarize the learning process in Algorithm 1.

During inference, we initialize the latent variables $\mathbf{z}$ for MCMC sampling from Gaussian white noise and run only the posterior sampling step to obtain $\mathbf{z}_+$. The inferred mask and region images are then given by the outputs of generative models $p_{\beta_k}(\mathbf{w}|\mathbf{z}_+)$ and $p_{\beta_k}(\mathbf{x}|\mathbf{z}_+),\ k = 1, 2$, respectively.

3.3 Technical Details

Pseudo Label for Additional Regularization Although the proposed Deep Region Competition algorithm explicitly models the interaction between the regions, it is still possible that the model converges to a trivial extractor, which treats the entire image as the foreground or background region, leaving the other region null. We exploit the symbolic vector $\mathbf{y}$ emitted by the LEBM (see Section 2) for additional regularization. The strategy is similar to the mutual information maximization used in InfoGAN [54]. In particular, we use the symbolic vector $\mathbf{y}$ inferred from $\mathbf{z}$ as the pseudo class label for $\mathbf{z}$ and train an auxiliary classifier jointly with the above models to ensure that the generated regions $\mathbf{x}_k,\ k = 1, 2$ contain similar symbolic information for $\mathbf{z}_k,\ k = 1, 2$. Intuitively, this loss can prevent the regions from converging to null since the symbolic representation $\mathbf{y}_k,\ k = 1, 2$ would never be well retrieved if that did happen.

Implementation We adopt a similar architecture for the generator as in DCGAN [55] throughout the experiments and only change the dimension of the latent variables $\mathbf{z}$ for different datasets. The generator consists of a fully connected layer followed by five stacked upsample-conv-norm layers. We replace the batch-norm layers [56] with instance-norm [57] in the architecture. The energy-term in LEBM is parameterized by a 3-layered MLP. We adopt orthogonal initialization [58] commonly used in generative models to initialize the networks and orthogonal regularization [35] to facilitate training. In addition, we observe performance improvement when adding Total-Variation norm [59] for the background generative model. More details, along with specifics of the implementation used in our experiments, are provided in the supplementary material.

4 Experiments

We design experiments to answer the following questions: (1) How does the proposed method compare to previous state-of-the-art competitors? (2) How do the proposed components contribute to the performance of the model? (3) Does the proposed method exhibit generalization on images containing unseen instances[1] and even objects from novel categories?

[1]The exact foreground objects are unseen during training, but similar objects from the same category are covered in the training data.

To answer these questions, we evaluate our method on five challenging datasets in two groups: (1) Caltech-UCSD Birds-200-2011 (Birds) [60], Stanford Dogs (Dogs) [61], and Stanford Cars (Cars) [62] datasets; (2) CLEVR6 [63] and Textured Multi-dSprites (TM-dSprites) [64] datasets. The first group of datasets covers complex real-world domains, whereas the second group features environments of multi-object foreground with challenging spatial configurations or confounding backgrounds. As to be shown, the proposed method is generic to various kinds of input and produces more competitive foreground-background partition results than previous methods.

4.1 Results on Foreground Extraction

Model	Single Object						Multi-Object			
	Birds		Dogs		Cars		CLEVR6		TM-dSprites	
	IoU	Dice	IoU	Dice	IoU	Dice	IoU	Dice	IoU	Dice
W-Net*	24.8	38.9	47.7	62.1	52.8	67.6	-	-	-	-
GrabCut	30.2	42.7	58.3	70.9	61.3	73.1	19.0	30.5	61.9	71.0
ReDO§	46.5	60.2	55.7	70.3	52.5	68.6	18.6	31.0	9.4	17.2
OneGAN*†	55.5	69.2	71.0	81.7	71.2	82.6	-	-	-	-
IODINE§	30.9	44.6	54.4	67.0	51.7	67.3	19.9	32.4	7.3	12.8
Slot-Attn.§	35.6	51.5	38.6	55.3	41.3	58.3	83.6	90.7	7.3	13.5
Ours	**56.4**	**70.9**	**71.7**	**83.2**	**72.4**	**83.7**	**84.7**	**91.5**	**78.8**	**87.5**

Table 1: Foreground extraction results measured in IoU and Dice. Higher is better in all scores. *Results of W-Net and OneGAN are provided by Benny and Wolf [30]. Note that results of these two models on Dogs and Cars datasets may not be directly comparable to other listed methods, as the data used for training and evaluation could be different. We include these results as a rough reference since no official implementation or pretrained model are publicly available. § indicates unfair baseline results obtained using extra ground-truth information, *i.e.*, we choose the best-matching scores from the permutation of foreground and background masks. †OneGAN is a strong **weakly supervised** baseline, which requires clean background images to provide additional supervision. We include this model as a virtual upper bound of the fully unsupervised methods.

Single Object in the Wild In the first group of datasets, there is typically a single object in the foreground, varying in shapes, texture, and lighting conditions. Unsupervised foreground extraction on these datasets requires much more sophisticated visual cues than colors and shapes. Birds dataset consists of 11,788 images of 200 classes of birds annotated with high-quality segmentation masks, Dogs dataset consists of 20,580 images of 120 classes annotated with bounding boxes, and Cars dataset consists of 16,185 images of 196 classes. The latter two datasets are primarily made for fine-grained categorization. To evaluate foreground extraction, we follow the practice in Benny and Wolf [30] and Savarese et al. [22], and approximate ground-truth masks for the images with Mask R-CNN [16], pre-trained on the MS COCO [9] dataset with a ResNet-101 [65] backend. The pre-trained model is acquired from the detectron2 [66] toolkit. This results in 5,024 dog images and 12,322 car images with a clear foreground-background setup and corresponding masks.

On datasets featuring a single foreground object, we use the 2-slot version of IODINE and Slot-attention. Since ReDO, IODINE, and Slot-Attention do not distinguish foreground and background in output regions, we choose the best-matching scores from the permutation of foreground and background masks as in [27]. We observe that the proposed method and Grabcut are the only two methods that provide explicit identification of foreground objects and background regions. While the Grabcut algorithm actually requires a predefined bounding box as input that specifies the foreground region, our method, thanks to the learned pixel re-assignment (see Section 3.1), can achieve this in a fully unsupervised manner. Results in Table 1 show that our method outperforms all the unsupervised baselines by a large margin, exhibiting comparable performance even to the weakly supervised baseline that requires additional background information as inputs [30]. We provide samples of foreground extraction results as well as generated background and foreground regions in Fig. 3. Note that our final goal is not to synthesize appealing images but to learn foreground extractors in a fully unsupervised manner. As the limitation of our method, DRC generates foreground and background regions less realistic than those generated by state-of-the-art GANs, which hints a possible direction for future work.

Multi-Object Scenes The second group of datasets contains images with possibly simpler foreground objects but more challenging scene configurations or background parts. Visual scenes in

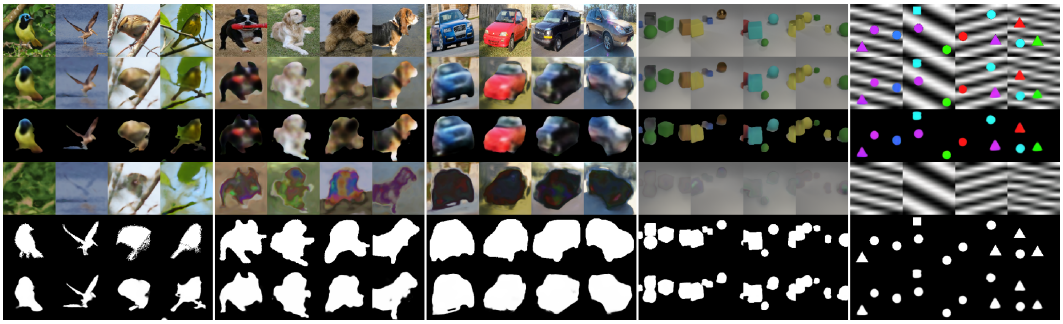

Figure 3: Foreground extraction results for each dataset (zoom in for better visibility). From top to bottom: (i) observed images, (ii) generated images, (iii) masked generated foregrounds, (iv) generated backgrounds, (v) ground-truth foreground masks, and (vi) inferred foreground masks. More samples and results of baselines can be found in the supplementary material.

the CLEVR6 dataset contain various objects and often with partial occlusions and truncations. Following the evaluation protocol in IODINE and Slot-attention, we use the first 70K samples from CLEVR [63] and filter the samples for scenes with at most 6 objects for training and evaluation, *i.e.*, CLEVR6. The TM-dSprites dataset is a variant of Multi-dSprites [64] but has strongly confounding backgrounds borrowed from Textured MNIST [34]. We generate 20K samples for the experiments. Similar to Greff et al. [32] and Locatello et al. [33], we evaluate on a subset containing 1K samples for testing. Note that IODINE and Slot-attention are designed for segmenting complex multi-object scenes using slot-based object representations. Ideally, the output of these models consists of masks for each individual object, while the background is viewed as a virtual "object" as well. In practice, however, it is possible that the model distributes the background over all the slots as mentioned in Locatello et al. [33]. We therefore propose two corresponding approaches (see the supplementary material for more details) to convert the output object masks into a foreground-background partition, and report the best results of these two options for IODINE and Slot-attention in Table 1.

On the CLEVR6 dataset, we use the publicly available pretrained model for IODINE, which achieves a reasonable ARI (excluding background pixels) of 94.4 on the testing data, close to the testing results in Greff et al. [32]. We observe that IODINE distributes the background over all the slots for some of the testing samples, resulting in much lower IoU and Dice scores. We re-train the Slot-attention model using the official implementation on CLEVR6, as no pretrained model is publicly available. The re-trained model achieves a foreground ARI of 98.0 on the 1K testing samples, which we consider as a sign of valid re-implementation. Results in Table 1 demonstrate that the proposed method can effectively process images of challenging multi-object scenes. To be specific, our method demonstrates competitive performance on the CLEVR6 dataset compared with the SOTA object discovery method. Moreover, as shown empirically in Fig. 3, the proposed method is able to handle the strongly confounding background introduced in Greff et al. [34], whereas previous methods are distracted by the background and mostly fail to capture the foreground objects.

4.2 Ablation Study

We provide ablation studies on the Birds dataset to inspect the contribution of each proposed component in our model. As shown in Table 2, we observe that replacing the LEBMs in the foreground and background models with amortized inference networks significantly harms the performance of the proposed method. In particular, the modified model fails to generate any meaningful results (marked as - in Table 2). We conjecture that LEBM benefits from the low-dimensionality of the latent space [46] and therefore enjoys more efficient learning. However, the inference networks need to learn an extra mapping from the high-dimensional image space to the latent space and require more elaborate architecture and tuning for convergence. Furthermore, we observe that the model that does not learn pixel re-assignment for background can still generate meaningful images, but only vaguely captures masks for foreground extraction.

Model	IoU	Dice
amortized inference*	-	-
w/o pix. re-assign.	21.8	35.3
w/o pseudo label	48.7	64.2
w/o TV-norm reg.	53.0	68.1
w/o ortho. reg.	54.3	69.2
short-run chain†	52.5	67.7
Full model	56.4	70.9

Table 2: Ablation study on Birds. *We replace the LEBM with encoders to perform amortized inference for the latent variables $\mathbf{z}$ within a variational framework as in VAE [67]. †We explore the possibility of using short-run MCMC [68] instead of persistent chain sampling.

4.3 Generalizable Foreground Extraction

Extracting Novel Foreground Objects from Training Categories We present results on generalizing to novel objects from the training classes. To evaluate our method, we split the Birds dataset following Chen et al. [27], resulting in 10K training images and 1K testing images. On Dogs and Cars datasets, we split the dataset based on the original train-test split [61, 62]. This split gives 3,286 dog images and 6,218 car images for training, and 1,738 dog images and 6,104 car images for testing, respectively. As summarized in Table 3, our method shows superior performances compared with baselines. The performance gap between training and testing is constantly small over all three datasets.

Model	Birds		Dogs		Cars	
	IoU	Dice	IoU	Dice	IoU	Dice
	Tr.|Te.	Tr.|Te.	Tr.|Te.	Tr.|Te.	Tr.|Te.	Tr.|Te.
GrabCut*	30.2|30.3	42.7|42.8	58.3|57.9	70.8|70.5	60.9|61.6	72.7|73.5
ReDO	46.8|47.1	61.4|61.7	54.3|52.8	69.2|67.9	52.6|52.5	68.7|68.6
Ours	**54.8|54.6**	**69.5|69.4**	**71.6|72.3**	**83.2|83.6**	**71.9|70.8**	**83.3|82.5**

Table 3: Performance of DRC on training and held-out testing data. *Note that GrabCut is a deterministic method that does not require training. We report the results of GrabCut [5] on these splits only for reference. Tr. indicates the performance on training data, and Te. indicates the performance on testing data.

Extracting Novel Foreground Objects from Unseen Categories To investigate how well our method generalizes to categories unseen during training, we evaluate the models trained in real-world single object datasets on the held-out testing data from different categories. We use the same training and testing splits on these datasets as in the previous experiments. Table 4 shows that our method outperforms the baselines on the Birds dataset when the model has trained on Dogs or Cars dataset, which have quite different distributions in foreground object shapes. Competitors like ReDO also exhibit such out-of-distribution generalization but only to a limited extent. Similar results are observed when using Dogs or Cars as the testing set. We can see that when the model is trained on Dogs and evaluated on Cars or vice versa, it still maintains comparable performances w.r.t. those are trained&tested on the same class. We hypothesize that these two datasets have similar distributions in foreground objects and background regions. In the light of this, we can further entail that the distribution of Dogs is most similar to that of Cars and less similar to that of Birds according to the results, which is consistent with our intuitive observation of the data. We provide a preliminary analysis of the statistics of these datasets in the supplementary material.

Test	Train	GrabCut		ReDO		Ours	
		IoU	Dice	IoU	Dice	IoU	Dice
Birds	Birds*			47.1	61.7	54.6	69.4
	Dogs	30.3	42.8	22.2	35.3	41.3	57.4
	Cars			16.4	27.7	39.2	55.3
Dogs	Dogs*			52.8	67.9	72.3	83.6
	Cars	57.9	70.5	44.5	61.2	67.8	80.4
	Birds			44.0	60.3	53.6	69.1
Cars	Cars*			52.5	68.6	70.8	82.5
	Dogs	61.6	73.5	51.6	67.1	68.6	81.0
	Birds			41.8	58.6	45.1	61.7

Table 4: Performance of DRC on unseen testing categories. *We include the testing results of models trained with data from the same categories for reference.

5 Conclusion

We have presented the Deep Region Competition, an EM-based fully unsupervised foreground extraction algorithm fueled by energy-based prior and generative image modeling. We propose learned pixel re-assignment as an inductive bias to capture the background regularities. Experiments demonstrate that DRC exhibits more competitive performances on complex real-world data and challenging multi-object scenes. We show empirically that learned pixel re-assignment helps to provide explicit identification for foreground and background regions. Moreover, we find that DRC can potentially generalize to novel foreground objects even from categories unseen during training. We hope our work will inspire future research along this challenging but rewarding research direction.

References

[1] Song Chun Zhu and Alan Yuille. Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 18(9):884–900, 1996.

[2] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 22(8):888–905, 2000.

[3] Zhuowen Tu and Song-Chun Zhu. Image segmentation by data-driven markov chain monte carlo. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 24(5):657–673, 2002.

[4] Yuri Y Boykov and M-P Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2001.

[5] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. " grabcut" interactive foreground extraction using iterated graph cuts. *ACM transactions on graphics (TOG)*, 23(3):309–314, 2004.

[6] Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, Philip HS Torr, and Shi-Min Hu. Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 37(3):569–582, 2014.

[7] Huaizu Jiang, Jingdong Wang, Zejian Yuan, Yang Wu, Nanning Zheng, and Shipeng Li. Salient object detection: A discriminative regional feature integration approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[8] Wangjiang Zhu, Shuang Liang, Yichen Wei, and Jian Sun. Saliency optimization from robust background detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[9] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2014.

[10] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision (IJCV)*, 88(2):303–338, 2010.

[11] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[12] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[13] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 39(12):2481–2495, 2017.

[14] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(4):834–848, 2017.

[15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 2015.

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2017.

[17] Nick Chater, Joshua B Tenenbaum, and Alan Yuille. Probabilistic models of cognition: Conceptual foundations. *Trends in Cognitive Sciences*, 10(7):287–291, 2006.

[18] Thomas F Shipley and Philip J Kellman. *From fragments to objects: Segmentation and grouping in vision*. Elsevier, 2001.

[19] Xide Xia and Brian Kulis. W-net: A deep model for fully unsupervised image segmentation. *arXiv preprint arXiv:1711.08506*, 2017.

[20] Asako Kanezaki. Unsupervised image segmentation by backpropagation. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[21] Xu Ji, João F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2019.

[22] Pedro Savarese, Sunnie SY Kim, Michael Maire, Greg Shakhnarovich, and David McAllester. Information-theoretic segmentation by inpainting error maximization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[23] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(5):898–916, 2010.

[24] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(11):2274–2282, 2012.

[25] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[26] Jianwei Yang, Anitha Kannan, Dhruv Batra, and Devi Parikh. Lr-gan: Layered recursive generative adversarial networks for image generation. In *International Conference on Learning Representations (ICLR)*, 2017.

[27] Mickaël Chen, Thierry Artières, and Ludovic Denoyer. Unsupervised object segmentation by redrawing. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[28] Pavel Ostyakov, Roman Suvorov, Elizaveta Logacheva, Oleg Khomenko, and Sergey I Nikolenko. Seigan: Towards compositional image generation by simultaneously learning to segment, enhance, and inpaint. *arXiv preprint arXiv:1811.07630*, 2018.

[29] Krishna Kumar Singh, Utkarsh Ojha, and Yong Jae Lee. Finegan: Unsupervised hierarchical disentanglement for fine-grained object generation and discovery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[30] Yaniv Benny and Lior Wolf. Onegan: Simultaneous unsupervised learning of conditional image generation, foreground segmentation, and fine-grained clustering. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020.

[31] Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019.

[32] Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *Proceedings of International Conference on Machine Learning (ICML)*, 2019.

[33] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[34] Klaus Greff, Antti Rasmus, Mathias Berglund, Tele Hotloo Hao, Jürgen Schmidhuber, and Harri Valpola. Tagger: Deep unsupervised perceptual grouping. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[35] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Neural photo editing with introspective adversarial networks. *arXiv preprint arXiv:1609.07093*, 2016.

[36] Sjoerd van Steenkiste, Michael Chang, Klaus Greff, and Jürgen Schmidhuber. Relational neural expectation maximization: Unsupervised discovery of objects and their interactions. In *International Conference on Learning Representations (ICLR)*, 2018.

[37] SM Ali Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Koray Kavukcuoglu, and Geoffrey E Hinton. Attend, infer, repeat: Fast scene understanding with generative models. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[38] Martin Engelcke, Adam R Kosiorek, Oiwi Parker Jones, and Ingmar Posner. Genesis: Generative scene inference and sampling with object-centric latent representations. In *International Conference on Learning Representations (ICLR)*, 2020.

[39] Zhixuan Lin, Yi-Fu Wu, Skand Vishwanath Peri, Weihao Sun, Gautam Singh, Fei Deng, Jindong Jiang, and Sungjin Ahn. Space: Unsupervised object-oriented scene representation via spatial attention and decomposition. In *International Conference on Learning Representations (ICLR)*, 2020.

[40] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988.

[41] Laurent D Cohen. On active contour models and balloons. *CVGIP: Image understanding*, 53 (2):211–218, 1991.

[42] Rolf Adams and Leanne Bischof. Seeded region growing. *IEEE Transactions on pattern analysis and machine intelligence*, 16(6):641–647, 1994.

[43] Song Chun Zhu, Yingnian Wu, and David Mumford. Filters, random fields and maximum entropy (frame): Towards a unified theory for texture modeling. *International Journal of Computer Vision (IJCV)*, 27(2):107–126, 1998.

[44] Cheng-en Guo, Song-Chun Zhu, and Ying Nian Wu. Primal sketch: Integrating structure and texture. *Computer Vision and Image Understanding (CVIU)*, 106(1):5–19, 2007.

[45] Yvan G Leclerc. Constructing simple stable descriptions for image partitioning. *International journal of computer vision*, 3(1):73–102, 1989.

[46] Bo Pang, Tian Han, Erik Nijkamp, Song-Chun Zhu, and Ying Nian Wu. Learning latent space energy-based prior model. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[47] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

[48] Michael I Jordan and Robert A Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 6(2):181–214, 1994.

[49] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of International Conference on Machine Learning (ICML)*, 2011.

[50] Radford M Neal et al. Mcmc using hamiltonian dynamics. In *Handbook of Markov Chain Monte Carlo*, page 2. 2011.

[51] Bo Pang, Erik Nijkamp, Jiali Cui, Tian Han, and Ying Nian Wu. Semi-supervised learning by latent space energy-based model of symbol-vector coupling. In *"I Can't Believe It's Not Better!"NeurIPS 2020 workshop*, 2020.

[52] Bela Julesz. Textons, the elements of texture perception, and their interactions. *Nature*, 290 (5802):91–97, 1981.

[53] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[54] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: interpretable representation learning by information maximizing generative adversarial nets. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[55] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[56] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of International Conference on Machine Learning (ICML)*, 2015.

[57] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[58] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

[59] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.

[60] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[61] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *CVPR Workshop on Fine-Grained Visual Categorization (FGVC)*, 2011.

[62] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV workshops*, 2013.

[63] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[64] Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

[65] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[66] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[67] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[68] Erik Nijkamp, Mitch Hill, Song-Chun Zhu, and Ying Nian Wu. Learning non-convergent non-persistent short-run mcmc toward energy-based model. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

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] See Section 4.1, Section 4.2 and Section 4.3.
   (b) Did you describe the limitations of your work? [Yes] See Section 4.1.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
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 provide pytorch-style codes and detailed instructions of how to reproduce the main results in the supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4.1, Section 4.2, Section 4.3 and supplemental material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (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] See supplemental material.
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]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] See Section 4.1, Section 4.2 and Section 4.3.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
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]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]