

Joint Training of Variational Auto-Encoder and Latent Energy-Based Model

Tian Han¹, Erik Nijkamp², Linqi Zhou², Bo Pang², Song-Chun Zhu², Ying Nian Wu²

¹Department of Computer Science, Stevens Institute of Technology

²Department of Statistics, University of California, Los Angeles

than6@stevens.edu, {enijkamp,bopang}@ucla.edu, linqizhou907@gmail.com

{sczhu,ywu}@stat.ucla.edu

Abstract

This paper proposes a joint training method to learn both the variational auto-encoder (VAE) and the latent energy-based model (EBM). The joint training of VAE and latent EBM are based on an objective function that consists of three Kullback-Leibler divergences between three joint distributions on the latent vector and the image, and the objective function is of an elegant symmetric and anti-symmetric form of divergence triangle that seamlessly integrates variational and adversarial learning. In this joint training scheme, the latent EBM serves as a critic of the generator model, while the generator model and the inference model in VAE serve as the approximate synthesis sampler and inference sampler of the latent EBM. Our experiments show that the joint training greatly improves the synthesis quality of the VAE. It also enables learning of an energy function that is capable of detecting out of sample examples for anomaly detection.

1. Introduction

The variational auto-encoder (VAE) [23, 35] is a powerful method for generative modeling and unsupervised learning. It consists of a generator model that transforms a noise vector to a signal such as image via a top-down convolutional network (also called deconvolutional network due to its top-down nature). It also consists of an inference model that infers the latent vector from the image via a bottom-up network. The VAE has seen many applications in image and video synthesis [12, 3] and unsupervised and semi-supervised learning [37, 22].

Despite its success, the VAE suffers from relatively weak synthesis quality compared to methods such as generative adversarial net (GANs) [11, 34] that are based on adversarial learning. While combing VAE objective function with the GAN objective function can improve the synthesis quality, such a combination is rather ad hoc. In this paper, we shall pursue a more systematic integration of variational

learning and adversarial learning. Specifically, instead of employing a discriminator as in GANs, we recruit a latent energy-based model (EBM) to mesh with VAE seamlessly in a joint training scheme.

The generator model in VAE is a directed model, with a known prior distribution on the latent vector, such as Gaussian white noise distribution, and a conditional distribution of the image given the latent vector. The advantage of such a model is that it can generate synthesized examples by direct ancestral sampling. The generator model defines a joint probability density of the latent vector and the image in a top-down scheme. We may call this joint density the generator density.

VAE also has an inference model, which defines the conditional distribution of the latent vector given the image. Together with the data distribution that generates the observed images, they define a joint probability density of the latent vector and the image in a bottom-up scheme. We may call this joint density the joint data density, where the latent vector may be considered as the missing data in the language of the EM algorithm [6].

As we shall explain later, the VAE amounts to joint minimization of the Kullback-Leibler divergence from data density to the generator density, where the joint minimization is over the parameters of both the generator model and the inference model. In this minimization, the generator density seeks to cover the modes of the data density, and as a result, the generator density can be overly dispersed. This may partially explain VAE's lack of synthesis quality.

Unlike the generator network, the latent EBM is an undirected model. It defines an unnormalized joint density on the latent vector and the image via a joint energy function. Such undirected form enables the latent EBM to better approximate the data density than the generator network. However, the maximum likelihood learning of latent EBM requires (1) inference sampling: sampling from the conditional density of the latent vector given the observed example and (2) synthesis sampling: sampling from the joint density of the latent vector and the image. Both inference

sampling and synthesis sampling require time-consuming Markov chain Monte Carlo (MCMC).

In this paper, we propose to jointly train the VAE and the latent EBM so that these two models can borrow strength from each other. The objective function of the joint training method consists of the Kullback-Leibler divergences between three joint densities of the latent vector and the image, namely the data density, the generator density, and the latent EBM density. The three Kullback-Leibler divergences form an elegant symmetric and anti-symmetric form of divergence triangle that integrates the variational learning and the adversarial learning seamlessly.

The joint training is beneficial to both the VAE and the latent EBM. The latent EBM has a more flexible form and can approximate the data density better than the generator model. It serves as a critic of the generator model by judging it against the data density. To the generator model, the latent EBM serves as a surrogate of the data density and a target density for the generator model to approximate. The generator model and the associated inference model, in return, serve as approximate synthesis sampler and inference sampler of the latent EBM, thus relieving the latent EBM of the burden of MCMC sampling.

Our experiments show that the joint training method can learn the generator model with strong synthesis ability. It can also learn an energy function that is capable of anomaly detection.

2. Contributions and related work

The following are contributions of our work. (1) We propose a joint training method to learn both VAE and latent EBM. The objective function is of a symmetric and anti-symmetric form of divergence triangle. (2) The proposed method integrates variational and adversarial learning. (3) The proposed method integrates the research themes initiated by the Boltzmann machine and Helmholtz machine.

The following are the themes that are related to our work.

(1) Variational and adversarial learning. Over the past few years, there has been an explosion in research on variational learning and adversarial learning, inspired by VAE [23, 35, 37, 12] and GAN [11, 34, 2, 41] respectively. One aim of our work is to find a natural integration of variational and adversarial learning. We also compare with some prominent methods along this theme in our experiments. Notably, adversarially learned inference (ALI) [9, 7] combines the learning of the generator model and inference model in an adversarial framework. It can be improved by adding conditional entropy regularization as in more recent methods ALICE [26] and SVAE [4]. Though these methods are trained using joint discriminator on image and latent vector, such a discriminator is not a probability density, thus it is not a latent EBM.

(2) Helmholtz machine and Boltzmann machine. Before

VAE and GAN took over, Boltzmann machine [1, 17, 36] and Helmholtz machine [16] are two classes of models for generative modeling and unsupervised learning. Boltzmann machine is the most prominent example of latent EBM. The learning consists of two phases. The positive phase samples from the conditional distribution of the latent variables given the observed example. The negative phase samples from the joint distribution of the latent variables and the image. The parameters are updated based on the differences between statistical properties of positive and negative phases. Helmholtz machine can be considered a precursor of VAE. It consists of a top-down generation model and the bottom-up recognition model. The learning also consists of two phases. The wake phase infers the latent variables based on the recognition model and updates the parameters of the generation model. The sleep phase generates synthesized data from the generation model and updates the parameters of the recognition model. Our work seeks to integrate the two classes of models.

(3) Divergence triangle for joint training of generator network and energy-based model. The generator and energy-based model can be trained separately using maximum likelihood criteria as in [13, 33], and they can also be trained jointly as recently explored by [20, 38, 14, 24]. In particular, [14] proposes a divergence triangle criterion for joint training. Our training criterion is also in the form of divergence triangle. However, the EBM in these papers is only defined on the image and there is no latent vector in the EBM. In our work, we employ latent EBM which defines a joint density on the latent vector and the image, thus this undirected joint density is more natural match to the generator density and the data density, both are directed joint densities of the latent vector and the image.

3. Models and learning

3.1. Generator network

Let z be the d -dimensional latent vector. Let x be the D -dimensional signal, such as an image. VAE consists of a generator model, which defines a joint probability density

$$p_{\theta}(x, z) = p(z)p_{\theta}(x|z), \quad (1)$$

in a top-down direction, where $p(z)$ is the known prior distribution of the latent vector z , such as uniform distribution or Gaussian white noise, i.e., $z \sim N(0, I_d)$, where I_d is the d -dimensional identity matrix. $p_{\theta}(x|z)$ is the conditional distribution of x given z . A typical form of $p_{\theta}(x|z)$ is such that $x = g_{\theta}(z) + \epsilon$, where $g_{\theta}(z)$ is parametrized by a top-down convolutional network (also called the deconvolutional network due to the top-down direction), with θ collecting all the weight and bias terms of the network. ϵ is the residual noise, and usually it is assumed $\epsilon \sim N(0, \sigma^2 I_D)$.

The generator network is a directed model. We call $p_\theta(x, z)$ the generator density. x can be sampled directly by first sampling z and then sampling x given z . This is sometimes called ancestral sampling in the literature [30].

The marginal distribution $p_\theta(x) = \int p_\theta(x, z) dz$. It is not in closed form. Thus the generator network is sometimes called the implicit generative model in the literature.

The inference of z can be based on the posterior distribution of z given x , i.e., $p_\theta(z|x) = p_\theta(x, z)/p_\theta(x)$. $p_\theta(z|x)$ is not in closed form.

3.2. Inference model

VAE assumes an inference model $q_\phi(z|x)$ with a separate set of parameters ϕ . One example of $q_\phi(z|x)$ is $N(\mu_\phi(x), V_\phi(x))$, where $\mu_\phi(x)$ is the d -dimensional mean vector, and $V_\phi(x)$ is the d -dimensional diagonal variance-covariance matrix. Both μ_ϕ and V_ϕ can be parametrized by bottom-up convolutional networks, whose parameters are denoted by ϕ . The inference model $q_\phi(z|x)$ is a closed form approximation to the true posterior $p_\theta(z|x)$.

3.3. Data density

Let $q_{\text{data}}(x)$ be the distribution that generates the observed images. In practice, expectation with respect to $q_{\text{data}}(x)$ can be approximated by the average over the observed training examples.

The reason we use the notation q to denote the data distribution $q_{\text{data}}(x)$ is that $q_{\text{data}}(x)$ can be naturally combined with the inference model $q_\phi(z|x)$, so that we have the joint density

$$q_\phi(x, z) = q_{\text{data}}(x)q_\phi(z|x). \quad (2)$$

The above is also a directional density in that it can be factorized in a bottom-up direction. We may call the joint density $q_\phi(x, z)$ the data density, where in the terminology of the EM algorithm, we may consider z as the missing data, and $q_\phi(z|x)$ as the imputation model of the missing data.

3.4. VAE

The top-down generator density $p_\theta(x, z) = p(z)p_\theta(x|z)$ and the bottom-up data density $q_\phi(x, z) = q_{\text{data}}(x)q_\phi(z|x)$ form a natural pair. As noted by [14], VAE can be viewed as the following joint minimization

$$\min_{\theta} \min_{\phi} \text{KL}(q_\phi(x, z) \| p_\theta(x, z)), \quad (3)$$

where for two densities $q(x)$ and $p(x)$ in general, $\text{KL}(q(x) \| p(x)) = \mathbb{E}_q[\log(q(x)/p(x))]$ is the Kullback-Leibler divergence between q and p .

To connect the above joint minimization to the usual form of VAE,

$$\begin{aligned} \text{KL}(q_\phi(x, z) \| p_\theta(x, z)) &= \text{KL}(q_{\text{data}}(x) \| p_\theta(x)) \\ &+ \mathbb{E}_{q_{\text{data}}(x)}[\text{KL}(q_\phi(z|x) \| p_\theta(z|x))], \end{aligned} \quad (4)$$

$$(5)$$

where for two joint densities $q(x, y)$ and $p(x, y)$, we define $\text{KL}(q(y|x) \| p(y|x)) = \mathbb{E}_{q(x|y)}[\log(q(y|x)/p(y|x))] \geq 0$.

Since $\text{KL}(q_{\text{data}}(x) \| p_\theta(x)) = \mathbb{E}_{q_{\text{data}}(x)}[\log q_{\text{data}}(x)] - \mathbb{E}_{q_{\text{data}}(x)}[\log p_\theta(x)]$, the joint minimization problem is equivalent to the joint maximization of

$$\begin{aligned} &\mathbb{E}_{q_{\text{data}}(x)}[\log p_\theta(x) - \text{KL}(q_\phi(z|x) \| p_\theta(z|x))] \\ &= \mathbb{E}_{q_{\text{data}}(x)}[\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x, z)] - \mathbb{E}_{q_\phi(z|x)}[\log q_\phi(z|x)]], \end{aligned} \quad (6)$$

which is the lower bound of the log-likelihood used in VAE [23].

It is worth of noting that the wake-sleep algorithm [16] for training the Helmholtz machine consists of (1) wake phase: $\min_{\theta} \text{KL}(q_\phi(x, z) \| p_\theta(x, z))$, and (2) sleep phase: $\min_{\phi} \text{KL}(p_\theta(x, z) \| q_\phi(x, z))$. The sleep phase reverses the order of KL-divergence.

3.5. Latent EBM

Unlike the directed joint densities $p_\theta(x, z) = p(z)p_\theta(x|z)$ in the generator network, and $q_\phi(x, z) = q_{\text{data}}(x)q_\phi(z|x)$ in the data density, the latent EBM defines an undirected joint density, albeit an unnormalized one:

$$\pi_\alpha(x, z) = \frac{1}{Z(\alpha)} \exp[f_\alpha(x, z)], \quad (7)$$

where $-f_\alpha(x, z)$ is the energy function (a term originated from statistical physics) defined on the image x and the latent vector z . $Z(\alpha) = \int \int \exp[f_\alpha(x, z)] dx dz$ is the normalizing constant. It is usually intractable, and $\exp[f_\alpha(x, z)]$ is an unnormalized density. The most prominent example of latent EBM is the Boltzmann machine [1, 17], where $f_\alpha(x, z)$ consists of pairwise potentials. In our work, we first encode x into a vector and then concatenate this vector with the vector z , and then get $f_\alpha(x, z)$ by a network defined on the concatenated vector.

3.6. Inference and synthesis sampling

Let $\pi_\alpha(x) = \int \pi_\alpha(x, z) dz$ be the marginal density of latent EBM. The maximum likelihood learning of α is based on $\min_{\alpha} \text{KL}(q_{\text{data}}(x) \| \pi_\alpha(x))$ because minimizing $\text{KL}(q_{\text{data}}(x) \| \pi_\alpha(x))$ is equivalent to maximizing the log-likelihood $\mathbb{E}_{q_{\text{data}}(x)}[\log \pi_\alpha(x)]$. The learning gradient is

$$\begin{aligned} \frac{\partial}{\partial \alpha} \mathbb{E}_{q_{\text{data}}(x)}[\log \pi_\alpha(x)] &= \mathbb{E}_{q_{\text{data}}(x)\pi_\alpha(z|x)} \left[\frac{\partial}{\partial \alpha} f_\alpha(x, z) \right] \\ &- \mathbb{E}_{\pi_\alpha(x, z)} \left[\frac{\partial}{\partial \alpha} f_\alpha(x, z) \right]. \end{aligned} \quad (8)$$

This is a well known result in latent EBM [1, 25]. On the right hand of the above equation, the two expectations can be approximated by Monte Carlo sampling. For each observed image, sampling from $\pi_\alpha(z|x)$ is to infer z from x . We call it the inference sampling. In the literature, it is

called the positive phase [1, 17]. It is also called clamped sampling where x is an observed image and is fixed. Sampling from $\pi_\alpha(x, z)$ is to generate synthesized examples from the model. We call it the synthesis sampling. In the literature, it is called the negative phase. It is also called unclamped sampling where x is also generated from the model.

4. Joint training

4.1. Objective function of joint training

We have the following three joint densities.

- (1) The generator density $p_\theta(x, z) = p(z)p_\theta(x|z)$.
- (2) The data density $q_\phi(x, z) = q_{\text{data}}(x)q_\phi(z|x)$.
- (3) The latent EBM density $\pi_\alpha(x, z)$.

We propose to learn the generator model parametrized by θ , the inference model parametrized by ϕ , and the latent EBM parametrized by α by the following divergence triangle:

$$\min_{\theta} \min_{\phi} \max_{\alpha} L(\theta, \alpha, \phi), \quad (9)$$

$$L = \text{KL}(q_\phi \| p_\theta) + \text{KL}(p_\theta \| \pi_\alpha) - \text{KL}(q_\phi \| \pi_\alpha), \quad (10)$$

where all the densities q_ϕ , p_θ , and π_α are joint densities of (x, z) .

The above objective function is in an symmetric and anti-symmetric form. The anti-symmetry is caused by the negative sign in front of $\text{KL}(q_\phi \| \pi_\alpha)$ and the maximization over α .

4.2. Learning of latent EBM

For learning the latent EBM, $\max_{\alpha} L$ is equivalent to minimizing

$$L_E(\alpha) = \text{KL}(q_\phi \| \pi_\alpha) - \text{KL}(p_\theta \| \pi_\alpha). \quad (11)$$

In the above minimization, π_α seeks to get close to the data density q_ϕ and get away from p_θ . Thus π_α serves as a critic of p_θ by comparing p_θ against q_ϕ . Because of the undirected form of π_α , it can be more flexible than the directional p_θ in approximating q_ϕ .

The gradient of the above minimization is

$$\begin{aligned} \frac{d}{d\alpha} L_E(\alpha) &= -\mathbb{E}_{q_{\text{data}}(x)q_\phi(z|x)} \left[\frac{\partial}{\partial \alpha} f_\alpha(x, z) \right] \\ &+ \mathbb{E}_{p_\theta(x, z)} \left[\frac{\partial}{\partial \alpha} f_\alpha(x, z) \right]. \end{aligned} \quad (12)$$

Comparing Eqn.12 to Eqn.8, we replace $\pi_\alpha(z|x)$ by $q_\phi(z|x)$ in the inference sampling in the positive phase, and we replace $\pi_\alpha(x, z)$ by $p_\theta(x, z)$ in the synthesis sampling in the negative phase. Both $q_\phi(z|x)$ and $p_\theta(x, z)$ can be sampled directly. Thus the joint training enables the latent

EBM to avoid MCMC in both inference sampling and synthesis sampling. In other words, the inference model serves as an approximate inference sampler for latent EBM, and the generator network serves as an approximate synthesis sampler for latent EBM.

4.3. Learning of generator network

For learning the generator network, $\min_{\theta} L$ is equivalent to minimizing

$$L_G(\theta) = \text{KL}(q_\phi \| p_\theta) + \text{KL}(p_\theta \| \pi_\alpha). \quad (13)$$

where the gradient can be computed as:

$$\begin{aligned} \frac{d}{d\theta} L_G(\theta) &= -\mathbb{E}_{q_{\text{data}}(x)q_\phi(z|x)} \left[\frac{\partial}{\partial \theta} \log p_\theta(x|z) \right] \\ &- \frac{\partial}{\partial \theta} \mathbb{E}_{p_\theta(x, z)} [f_\alpha(x, z)]. \end{aligned} \quad (14)$$

In $\text{KL}(q_\phi \| p_\theta)$, p_θ appears on the right hand side of KL-divergence. Minimizing this KL-divergence with respect to θ requires p_θ to cover all the major modes of q_ϕ . If p_θ is not flexible enough, it will strain itself to cover all the major modes, and as a result, it will make p_θ over-dispersed than q_ϕ . This may be the reason that VAE tends to suffer in synthesis quality.

However, in the second term, $\text{KL}(p_\theta \| \pi_\alpha)$, p_θ appears on the left hand side of KL-divergence, and π_α , which seeks to get close to q_ϕ and get away from p_θ in its dynamics, serves as a surrogate for data density q_ϕ , and a target for p_θ . Because p_θ appears on the left hand side of KL-divergence, it has the mode chasing behavior, i.e., it may chase some major modes of π_α (surrogate of q_ϕ), while it does not need to cover all the modes. Also note that in $\text{KL}(p_\theta \| \pi_\alpha)$, we do not need to know $Z(\alpha)$ because it is a constant as far as θ is concerned.

Combine the above two KL-divergences, approximately, we minimize a symmetrized version of KL-divergence $S(q_\phi \| p_\theta) = \text{KL}(q_\phi \| p_\theta) + \text{KL}(p_\theta \| q_\phi)$ (assuming π_α is close to q_ϕ). This will correct the over-dispersion of VAE, and improve the synthesis quality of VAE.

We refer the reader to the textbook [10, 31] on the difference between $\min_p \text{KL}(q \| p)$ and $\min_p \text{KL}(p \| q)$. In the literature, they are also referred to as inclusive and exclusive KL, or KL and reverse KL.

4.4. An adversarial chasing game

The dynamics of π_α is that it seeks to get close to the data density q_ϕ and get away from p_θ . But the dynamics of p_θ is that it seeks to get close to π_α (and at the same time also get close to the data density q_ϕ). This defines an adversarial chasing game, i.e., π_α runs toward q_ϕ and runs from p_θ , while p_θ chases π_α . As a result, π_α leads p_θ toward q_ϕ . p_θ and π_α form an actor-critic pair.

4.5. Learning of inference model

The learning of the inference model $q_\phi(z|x)$ can be based on $\min_\phi L(\theta, \alpha, \phi)$, which is equivalent to minimizing

$$L_I(\phi) = \text{KL}(q_\phi \| p_\theta) - \text{KL}(q_\phi \| \pi_\alpha). \quad (15)$$

$q_\phi(z|x)$ seeks to be close to $p_\theta(z|x)$ relative to $\pi_\alpha(z|x)$. That is, $q_\phi(z|x)$ seeks to be the inference model for p_θ . Meanwhile, $\pi_\alpha(z|x)$ seeks to be close to $q_\phi(z|x)$. This is also a chasing game. $q_\phi(z|x)$ leads $\pi_\alpha(z|x)$ to be close to $p_\theta(z|x)$.

The gradient of $L_I(\phi)$ in Eqn.15 can be readily computed as:

$$\begin{aligned} \frac{d}{d\phi} L_I(\phi) &= \frac{\partial}{\partial \phi} \mathbb{E}_{q_\phi(x,z)} [\log q_\phi(x,z) - \log p_\theta(x,z)] \\ &\quad - \frac{\partial}{\partial \phi} \mathbb{E}_{q_\phi(x,z)} [\log q_\phi(x,z) - f_\alpha(x,z)]. \end{aligned} \quad (16)$$

We may also learn $q_\phi(z|x)$ by minimizing

$$L_I(\phi) = \text{KL}(q_\phi \| p_\theta) + \text{KL}(q_\phi \| \pi_\alpha), \quad (17)$$

where we let $q_\phi(z|x)$ to be close to both $p_\theta(z|x)$ and $\pi_\alpha(z|x)$ in variational approximation.

4.6. Algorithm

Algorithm 1 Joint training for VAE and latent EBM

Require:

training images $\{x_i\}_{i=1}^n$; number of learning iterations T ; $\alpha, \theta, \phi \leftarrow$ initialized network parameters.

Ensure:

estimated parameters $\{\alpha, \theta, \phi\}$; generated samples $\{\tilde{x}_i\}_{i=1}^{\tilde{M}}$.

- 1: Let $t \leftarrow 0$.
 - 2: **repeat**
 - 3: **Synthesis sampling for** $(z_i, \tilde{x}_i)_{i=1}^{\tilde{M}}$ using Eqn.18.
 - 4: **Inference sampling for** $(x_i, \tilde{z}_i)_{i=1}^{\tilde{M}}$ using Eqn.18.
 - 5: **Learn latent EBM:** Given $\{z_i, \tilde{x}_i\}_{i=1}^{\tilde{M}}$ and $\{x_i, \tilde{z}_i\}_{i=1}^{\tilde{M}}$, update $\alpha \leftarrow \alpha - \eta_\alpha L'_E(\alpha)$ using Eqn. 19 with learning rate η_α .
 - 6: **Learn inference model:** Given $\{x_i, \tilde{z}_i\}_{i=1}^{\tilde{M}}$, update $\phi \leftarrow \phi - \eta_\phi L'_I(\phi)$, with learning rate η_ϕ using Eqn. 20.
 - 7: **Learn generator network:** Given $\{z_i, \tilde{x}_i\}_{i=1}^{\tilde{M}}$ and $\{x_i, \tilde{z}_i\}_{i=1}^{\tilde{M}}$, update $\theta \leftarrow \theta - \eta_\theta L'_G(\theta)$, with learning rate η_θ using Eqn. 21.
 - 8: Let $t \leftarrow t + 1$.
 - 9: **until** $t = T$
-

The latent EBM, generator and inference model can be jointly trained using stochastic gradient descent based on

Eqn.12, Eqn.14 and Eqn.16. In practice, we use sample average to approximate the expectation.

Synthesis and inference sampling. The expectations for gradient computation are based on generator density $p_\theta(x, z)$ and data density $q_\phi(x, z)$. To approximate the expectation of generator density $\mathbb{E}_{p_\theta(x,z)}[\cdot]$, we perform synthesis sampling through $z \sim p(z)$, $\tilde{x} \sim p_\theta(x|z)$ to get \tilde{M} samples (z_i, \tilde{x}_i) . To approximate the expectation of data density $\mathbb{E}_{q_\phi(x,z)}[\cdot]$, we perform inference sampling through $x \sim q_{\text{data}}(x)$, $\tilde{z} \sim q_\phi(z|x)$ to get \tilde{M} samples (x_i, \tilde{z}_i) . Both $p_\theta(x|z)$ and $q_\phi(z|x)$ are assumed to be Gaussian, therefore we have:

$$\begin{aligned} \tilde{x} &= g_\theta(z) + \sigma e_1, \quad e_1 \sim \mathcal{N}(0, I_D); \\ \tilde{z} &= \mu_\phi(x) + V_\phi(x)^{1/2} e_2, \quad e_2 \sim \mathcal{N}(0, I_d), \end{aligned} \quad (18)$$

where $g_\theta(z)$ is the top-down deconvolutional network for the generator model (see Sec 3.1), and $\mu_\phi(x)$ and $V_\phi(x)$ are bottom-up convolutional networks for the mean vector and the diagonal variance-covariance matrix of the inference model (see Sec 3.2). We follow the common practice [11] to have \tilde{x} directly from generator network, i.e., $\tilde{x} \approx g_\theta(z)$. Note that the synthesis sample (z, \tilde{x}) and the inference sample (x, \tilde{z}) are functions of the generator parameter θ and the inference parameter ϕ respectively which ensure gradient back-propagation.

Model learning. The obtained synthesis samples and inference samples can be used to approximate the expectations in model learning. Specifically, for latent EBM learning, the gradient in Eqn.12 can be approximated by:

$$\begin{aligned} \frac{d}{d\alpha} L_E(\alpha) &\approx -\frac{1}{M} \sum_{i=1}^M \left[\frac{\partial}{\partial \alpha} f_\alpha(x_i, \tilde{z}_i) \right] \\ &\quad + \frac{1}{\tilde{M}} \sum_{i=1}^{\tilde{M}} \left[\frac{\partial}{\partial \alpha} f_\alpha(\tilde{x}_i, z_i) \right]. \end{aligned} \quad (19)$$

For inference model, the gradient in Eqn.16 can be approximated by:

$$\begin{aligned} \frac{d}{d\phi} L_I(\phi) &\approx \frac{\partial}{\partial \phi} \frac{1}{M} \sum_{i=1}^M [\log q_\phi(x_i, \tilde{z}_i) - \log p_\theta(x_i, \tilde{z}_i)] \\ &\quad - \frac{\partial}{\partial \phi} \frac{1}{\tilde{M}} \sum_{i=1}^{\tilde{M}} [\log q_\phi(x_i, \tilde{z}_i) - f_\alpha(x_i, \tilde{z}_i)]. \end{aligned} \quad (20)$$

For generator model, the gradient in Eqn.14 can be approximated by:

$$\begin{aligned} \frac{d}{d\theta} L_G(\theta) &\approx -\frac{1}{M} \sum_{i=1}^M \left[\frac{\partial}{\partial \theta} \log p_\theta(x_i | \tilde{z}_i) \right] \\ &\quad - \frac{\partial}{\partial \theta} \frac{1}{\tilde{M}} \sum_{i=1}^{\tilde{M}} [f_\alpha(\tilde{x}_i, z_i)]. \end{aligned} \quad (21)$$

Notice that the gradients in Eqn.20 and Eqn.21 on synthesis samples (z_i, \tilde{x}_i) and inference samples (x_i, \tilde{z}_i) can be easily back-propagated using Eqn.18. The detailed training procedure is presented in Algorithm 1.

5. Experiments

In this section, we evaluate the proposed model on four tasks: image generation, test image reconstruction, out-of-distribution generalization, and anomaly detection. The learning of inference model is based on Eqn.15, and we also tested the alternative way to train inference model using Eqn.17 for generation and reconstruction. We mainly consider 4 datasets which includes CIFAR-10, CelebA [27], Large-scale Scene Understand (LSUN) dataset [39] and MNIST. We will describe the datasets in more detail in the following relevant subsections. All the training image datasets are resized and scaled to $[-1, 1]$ with no further pre-processing. All network parameters are initialized with zero-mean gaussian with standard deviation 0.02 and optimized using Adam [21]. We adopt a similar deconvolutional network structure as in [34] for generator model and the “mirror” convolutional structure for inference model. Both structures involve batch normalization [19]. For joint energy model $\pi_\alpha(x, z)$, we use multiple layers of convolution to transform the observation x and the latent factor z , then concatenate them at the higher layer which shares similarity as in [26]. Spectral normalization is used as suggested in [29]. We refer to our project page¹ for details.

5.1. Image generation

In this experiment, we evaluate the visual quality of the generated samples. The well-learned generator network p_θ could generate samples that are realistic and share visual similarities as the training data. We mainly consider three common-used datasets including CIFAR-10, CelebA [27] and LSUN [39] for generation and reconstruction evaluation. The CIFAR-10 contains 60,000 color images of size 32×32 of which 50,000 images are for training and 10,000 are for testing. For CelebA dataset, we resize them to be 64×64 and randomly select 10,000 images of which 9,000 are for training and 1,000 are for testing. For LSUN dataset, we select the *bedroom* category which contains roughly 3 million images and resize them to 64×64 . We separate 10,000 images for testing and use the rest for training. The qualitative results are shown in Figure 1.

We further evaluate our model quantitatively by using Frechet Inception Distance (FID) [28] in Table 1. We compare with baseline models including VAE [23], DCGAN [34], WGAN [2], CoopNet [38], ALICE [26], SVAE [4] and SNGAN [29]. The FID scores are from the relevant papers, and for missing evaluations, we re-evaluate

them by utilizing their released codes or re-implement using similar structures and optimal parameters as indicated in their papers. From Table 1, our model achieves competitive generation performance compared to listed baseline models. Further compared to [14] which has 7.23 Inception Score (IS) on CIFAR-10 and 31.9 FID on CelebA, our model has 7.17 IS and 24.7 FID respectively. It can be shown that our joint training can greatly improve the synthesis quality compared to VAE alone. Note that the SNGAN [29] get better generation on CIFAR-10 which has relatively small resolution, while on other datasets that have relatively high resolution and diverse patterns, our model obtains more favorable results and has more stable training.

5.2. Testing image reconstruction

In this experiment, we evaluate the accuracy of the learned inference model by testing image reconstruction. The well-trained inference model should not only help to learn the latent EBM model but also learn to match the true posterior $p_\theta(z|x)$ of the generator model. Therefore, in practice, a well-learned inference model can be balanced to render both realistic generation, as we show in the previous section, as well as faithful reconstruction on testing images.

We evaluate the model on hold-out testing set of CIFAR-10, CelebA and LSUN-bedroom. Specifically, we use its own 10,000 testing images for CIFAR-10, 1,000 and 10,000 hold-out testing images for CelebA and LSUN-bedroom. The testing images and the corresponding reconstructions are shown in Figure 2. We also quantitatively compare with baseline models (ALI [9], ALICE [26], SVAE [4]) using Rooted Mean Square Error (RMSE). Note that for this experiment, we only compare with the relevant baseline models that contain joint discriminator on (x, z) and could achieve the decent generation quality. Besides, we do not consider the GANs and their variants because they have no inference model involved and are infeasible for image reconstruction. Table 2 shows the results. VAE is naturally integrated into our probabilistic model for joint learning. However, using VAE alone can be extremely ineffective on complex dataset. Our model instead achieves both the high generation quality as well as the accurate reconstruction.

5.3. Out-of-distribution generalization

In this experiment, we evaluate out-of-distribution (OOD) detection using the learned latent EBM $\pi_\alpha(x, z)$. If the energy model is well-learned, then the training image, together with its inferred latent factor, should form local energy minima. Unseen images from other distribution other than training ones should be assigned to relatively higher energies. This is closely related to the model of associative memory as observed by Hopfield [18].

We learn the proposed model on CIFAR-10 training im-

¹<https://hthth0801.github.io/jointLearning/>



Figure 1: Generated samples. Left: CIFAR-10 generation. Middle: CelebA generation. Right: LSUN bedroom generation.

Model	VAE	DCGAN	WGAN	CoopNet	ALICE	SVAE	SNGAN	Ours(+)	Ours
CIFAR-10	109.5	37.7	40.2	33.6	48.6	43.5	29.3	33.3	30.1
CelebA	99.09	38.4	36.4	56.6	46.1	40.7	50.4	29.5	24.7
LSUN	175.2	70.4	67.7	35.4	72	-	67.8	31.4	27.3

Table 1: Sample quality evaluation using FID scores on various datasets. Ours(+) denotes our proposed method with inference model trained using Eqn.17.

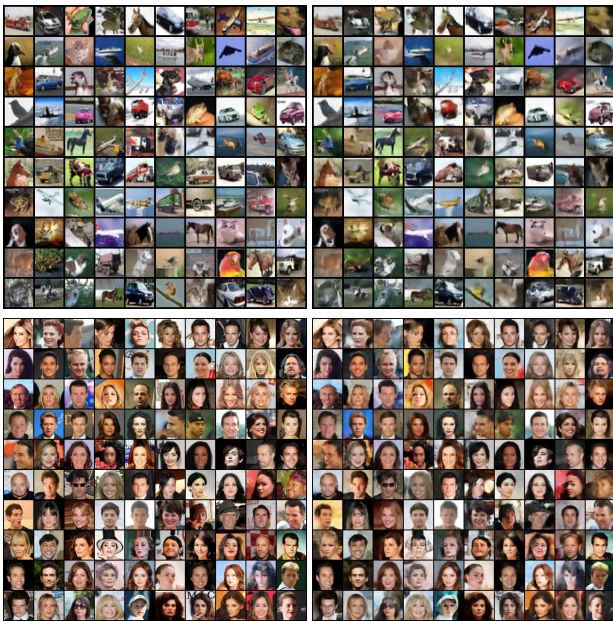


Figure 2: Test image reconstruction. Top: CIFAR-10. Bottom: CelebA. Left: test images. Right: reconstructed images.

Model	CIFAR-10	CelebA	LSUN-bedroom
VAE	0.192	0.197	0.164
ALI	0.558	0.720	-
ALICE	0.185	0.214	0.181
SVAE	0.258	0.209	-
Ours(+)	0.184	0.208	0.169
Ours	0.177	0.190	0.169

Table 2: Testing image reconstruction evaluation using RMSE. Ours(+) denotes our proposed method with inference model trained using Eqn.17.

ages, then utilize the learned energy model to classify the CIFAR-10 testing images from other OOD images using energy value (i.e., negative log likelihood). We use area under

the ROC curve (AUROC) scores as our OOD metric following [15] and we use Textures [5], uniform noise, SVHN [32] and CelebA images as OOD distributions (Figure 4 provides CIFAR-10 test images and examples of OOD images). We compare with ALICE [26], SVAE [4] and the recent EBM [8] as our baseline models. The CIFAR-10 training for ALICE, SVAE follow the network, hyperparameters and scores for EBM are taken directly from [8]. Table 3 shows the AUROC scores. We also provide histograms of relative likelihoods for OOD distributions in Figure 3 which can further verify that images from OOD distributions are assigned to relatively low log likelihood (i.e., high energy) compared to the training distribution. Our latent EBM could be learned to assign low energy for training distribution and high energy for data that comes from OOD distributions.

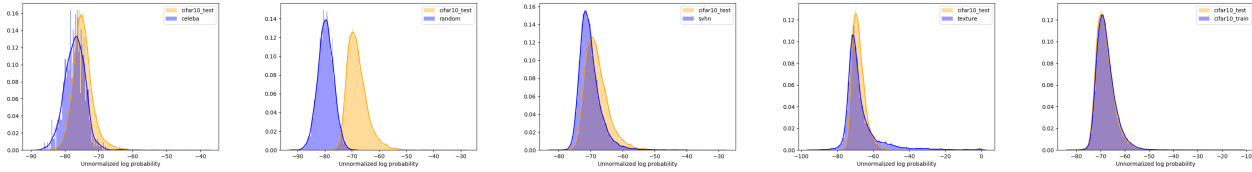


Figure 3: Histogram of log likelihood (unnormalized) for various datasets. We provide the histogram comparison between CIFAR-10 test set and CelebA, Uniform Random, SVHN, Texture and CIFAR-10 train set respectively.



Figure 4: Illustration of images from CIFAR-10 test, SVHN, Uniform Random, Texture and CelebA. The last four are considered to be OOD distributions.

Model	SVHN	Uniform	Texture	CelebA
EBM	0.63	1.0	0.48	-
ALICE	0.29	0.0	0.40	0.48
SVAE	0.42	0.29	0.5	0.52
Ours	0.68	1.0	0.56	0.56

Table 3: AUROC scores of OOD classification on various images datasets. All models are learned on CIFAR-10 train set.

5.4. Anomaly detection

In this experiment, we take a closer and more general view of the learned latent EBM with applications to anomaly detection. Unsupervised anomaly detection is one of the most important problems in machine learning and offers great potentials in many areas including cyber-security, medical analysis and surveillance etc. It is similar to the out-of-distribution detection discussed before, but can be more challenging in practice because the anomaly data may come from the distribution that is similar to and not entirely apart from the training distribution. We evaluate our model on MNIST benchmark dataset.

The MNIST dataset contains 60,000 gray-scale images of size 28×28 depicting handwritten digits. Following the same experiment setting as [24, 40], we make each digit class an anomaly and consider the remaining 9 digits as normal examples. Our model is trained with only normal data and tested with both normal and anomalous data. We use energy function as our decision function and compare with the BiGAN-based anomaly detection model [40], the recent MEG [24] and the VAE model using area under the

precision-recall curve (AUPRC) as in [40]. Table 4 shows the results.

Holdout	VAE	MEG	BiGAN- σ	Ours
1	0.063	0.281 ± 0.035	0.287 ± 0.023	0.297 ± 0.033
4	0.337	0.401 ± 0.061	0.443 ± 0.029	0.723 ± 0.042
5	0.325	0.402 ± 0.062	0.514 ± 0.029	0.676 ± 0.041
7	0.148	0.290 ± 0.040	0.347 ± 0.017	0.490 ± 0.041
9	0.104	0.342 ± 0.034	0.307 ± 0.028	0.383 ± 0.025

Table 4: AUPRC scores for unsupervised anomaly detection on MNIST. Numbers are taken from [24] and results for our model are averaged over last 10 epochs to account for variance.

6. Conclusion

This paper proposes a joint training method to learn both the VAE and the latent EBM simultaneously, where the VAE serves as an actor and the latent EBM serves as a critic. The objective function is of a simple and compact form of divergence triangle that involves three KL-divergences between three joint densities on the latent vector and the image. This objective function integrates both variational learning and adversarial learning. Our experiments show that the joint training improves the synthesis quality of VAE, and it learns reasonable energy function that is capable of anomaly detection.

Learning well-formed energy landscape remains a challenging problem, and our experience suggests that the learned energy function can be sensitive to the setting of hyper-parameters and within the training algorithm. In our further work, we shall further improve the learning of the energy function. We shall also explore joint training of models with multiple layers of latent variables in the styles of Helmholtz machine and Boltzmann machine.

Acknowledgment

The work is supported by DARPA XAI project N66001-17-2-4029; ARO project W911NF1810296; and ONR MURI project N00014-16-1-2007; and Extreme Science and Engineering Discovery Environment (XSEDE) grant ASC170063.

References

- [1] David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169, 1985.
- [2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223, 2017.
- [3] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H Campbell, and Sergey Levine. Stochastic variational video prediction. *arXiv preprint arXiv:1710.11252*, 2017.
- [4] Liqun Chen, Shuyang Dai, Yunchen Pu, Erjin Zhou, Chunyuan Li, Qinliang Su, Changyou Chen, and Lawrence Carin. Symmetric variational autoencoder and connections to adversarial learning. In *International Conference on Artificial Intelligence and Statistics*, pages 661–669, 2018.
- [5] Mircea Cimpoi, Subhansu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2014.
- [6] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.
- [7] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.
- [8] Yilun Du and Igor Mordatch. Implicit generation and generalization in energy-based models. *arXiv preprint arXiv:1903.08689*, 2019.
- [9] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.
- [10] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
- [11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.
- [12] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015.
- [13] Tian Han, Yang Lu, Song-Chun Zhu, and Ying Nian Wu. Alternating back-propagation for generator network. In *AAAI*, volume 3, page 13, 2017.
- [14] Tian Han, Erik Nijkamp, Xiaolin Fang, Mitch Hill, Song-Chun Zhu, and Ying Nian Wu. Divergence triangle for joint training of generator model, energy-based model, and inferential model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8670–8679, 2019.
- [15] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.
- [16] Geoffrey E Hinton, Peter Dayan, Brendan J Frey, and Radford M Neal. The “wake-sleep” algorithm for unsupervised neural networks. *Science*, 268(5214):1158–1161, 1995.
- [17] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.
- [18] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982.
- [19] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.
- [20] Taesup Kim and Yoshua Bengio. Deep directed generative models with energy-based probability estimation. *arXiv preprint arXiv:1606.03439*, 2016.
- [21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [22] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pages 3581–3589, 2014.
- [23] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [24] Rithesh Kumar, Anirudh Goyal, Aaron Courville, and Yoshua Bengio. Maximum entropy generators for energy-based models. *arXiv preprint arXiv:1901.08508*, 2019.
- [25] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.
- [26] Chunyuan Li, Hao Liu, Changyou Chen, Yuchen Pu, Liqun Chen, Ricardo Henao, and Lawrence Carin. Alice: Towards understanding adversarial learning for joint distribution matching. In *Advances in Neural Information Processing Systems*, pages 5495–5503, 2017.
- [27] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.
- [28] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are gans created equal? a large-scale study. *arXiv preprint arXiv:1711.10337*, 2017.
- [29] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.
- [30] Shakir Mohamed and Balaji Lakshminarayanan. Learning in implicit generative models. *arXiv preprint arXiv:1610.03483*, 2016.
- [31] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.
- [32] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bisacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.
- [33] Erik Nijkamp, Mitch Hill, Tian Han, Song-Chun Zhu, and Ying Nian Wu. On the anatomy of mcmc-based maximum likelihood learning of energy-based models. *arXiv preprint arXiv:1903.12370*, 2019.

- [34] Alec Radford, Luke Metz, and Soumith Chintala. Un-supervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.
- [35] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.
- [36] Ruslan Salakhutdinov and Hugo Larochelle. Efficient learning of deep boltzmann machines. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 693–700, 2010.
- [37] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. In *Advances in neural information processing systems*, pages 3738–3746, 2016.
- [38] Jianwen Xie, Yang Lu, Ruiqi Gao, Song-Chun Zhu, and Ying Nian Wu. Cooperative training of descriptor and generator networks. *arXiv preprint arXiv:1609.09408*, 2016.
- [39] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.
- [40] Houssam Zenati, Chuan Sheng Foo, Bruno Lecouat, Gaurav Manek, and Vijay Ramaseshan Chandrasekhar. Efficient gan-based anomaly detection. *arXiv preprint arXiv:1802.06222*, 2018.
- [41] Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016.