Divergence Triangle for Joint Training of Generator Model, Energy-based Model, and Inferential Model

Tian Han[1]*, Erik Nijkamp[1]*, Xiaolin Fang[2], Mitch Hill[1], Song-Chun Zhu[1], Ying Nian Wu[1]
[1]University of California, Los Angeles [2]Zhejiang University
{hantian, enijkamp, mkhil}@ucla.edu, xiaolinfang@zju.edu.cn, {sczhu,ywu}@stat.ucla.edu

Abstract

This paper proposes the divergence triangle as a framework for joint training of a generator model, energy-based model and inference model. The divergence triangle is a compact and symmetric (anti-symmetric) objective function that seamlessly integrates variational learning, adversarial learning, wake-sleep algorithm, and contrastive divergence in a unified probabilistic formulation. This unification makes the processes of sampling, inference, and energy evaluation readily available without the need for costly Markov chain Monte Carlo methods. Our experiments demonstrate that the divergence triangle is capable of learning (1) an energy-based model with well-formed energy landscape, (2) direct sampling in the form of a generator network, and (3) feed-forward inference that faithfully reconstructs observed as well as synthesized data.

1. Introduction

1.1. Integrating Three Models

Deep probabilistic generative models are a powerful framework for representing complex data distributions. They have been widely used in unsupervised learning problems to learn from unlabeled data. The goal of generative learning is to build rich and flexible models to fit complex, multi-modal data distributions as well as to be able to generate samples with high realism. The family of generative models may be roughly divided into two classes: The first class is the *energy-based model* (a.k.a undirected graphical model) and the second class is the latent variable model (a.k.a directed graphical model) which usually includes *generator model* for the generation and *inference model* for inference or reconstruction.

These models have their advantages and limitations. An energy-based model defines an explicit likelihood of the observed data up to a normalizing constant. However, sampling from such a model usually requires expensive Markov chain Monte Carlo (MCMC). A generator model defines direct sampling of the data. However, it does not have an explicit likelihood. The inference of the latent variables also requires MCMC sampling from the posterior distribution. The inference model defines an explicit approximation to the posterior distribution of the latent variables.

Combining the energy-based model, the generator model, and the inference model to get the best of each model is an attractive goal. On the other hand, challenges may accumulate when the models are trained together since different models need to effectively compete or cooperate together to achieve their highest performances. In this work, we propose the divergence triangle for joint training of energy-based model, generator model and inference model. The learning of three models can then be seamlessly integrated in a principled probabilistic framework. The energy-based model is learned based on the samples supplied by the generator model. With the help of the inference model, the generator model is trained by both the observed data and the energy-based model. The inference model is learned from both the real data fitted by the generator model as well as the synthesized data generated by the generator model.

Our experiments demonstrate that the divergence triangle is capable of learning an energy-based model with a well-behaved energy landscape, a generator model with highly realistic samples, and an inference model with faithful reconstruction ability.

1.2. Prior Art

The maximum likelihood learning of the energy-based model requires expectation with respect to the current model, while the maximum likelihood learning of the generator model requires expectation with respect to the posterior distribution of the latent variables. Both expectations can be approximated by MCMC, such as Gibbs sampling [11], Langevin dynamics, or Hamiltonian Monte Carlo (HMC) [34]. [31, 48] used Langevin dynamics for learning the energy-based models, and [13] used Langevin dynamics for learning the generator model. In both cases,

*Equal contributions.

MCMC sampling introduces an inner loop in the training procedure, posing a computational expense.

An early version of the energy-based model is the FRAME (Filters, Random field, And Maximum Entropy) model [53, 45]. [52] used gradient-based method such as Langevin dynamics to sample from the model. [51] called the energy-based models as descriptive models. [31, 48] generalized the model to deep variants.

Contrastive divergence (CD) [15] initializes finite step MCMC from the observed data to reduce the computational cost of sampling when learning an energy-based model [28]. The resulting learning algorithm follows the gradient of the difference between two Kullback-Leibler divergences, thus the name contrastive divergence. In this paper, we shall use the term "contrastive divergence" in a more general sense than [15]. Persistent contrastive divergence [42] initializes MCMC sampling from the samples of the previous learning iteration.

Generalizing [43], [21] developed an introspective learning method where the energy function is discriminatively learned, and the energy-based model is both a generative model and a discriminative model.

For learning the generator model, the variational auto-encoder (VAE) [25, 38, 33] approximates the posterior distribution of the latent variables by an explicit inference model. In VAE, the inference model is learned jointly with the generator model from the observed data. A precursor of VAE is the wake-sleep algorithm [17], where the inference model is learned from the dream data generated by the generator model in the sleep phase.

The generator model can also be learned jointly with a discriminator model, as in the generative adversarial networks (GAN) [12], as well as deep convolutional GAN (DCGAN) [37], energy-based GAN (EB-GAN) [50], Wasserstein GAN (WGAN) [2]. GAN does not involve an inference model.

The generator model can also be learned jointly with an energy-based model [23, 6]. We can interpret the learning scheme as an adversarial version of contrastive divergence. In GAN, the discriminator model eventually becomes confused between real and fake images, while in the joint learning of the generator model and the energy-based model, the learned energy-based model becomes a well-defined probability distribution on the observed data. The joint learning bears some similarity to WGAN, but unlike WGAN, the learning framework involves two complementary probability distributions.

The cooperative learning method of [47] bridges the gap between the energy-based model and generator model by initializing finite-step MCMC sampling of the energy-based model from images synthesized by the generator model. Such finite-step MCMC produces revised samples that closer to the modes of the energy-based model, and the generator model can learn from the MCMC revisions of its initial samples.

Adversarially learned inference (ALI) [10, 9] combines the learning of the generator model and inference model in an adversarial framework. ALI can be improved by adding conditional entropy regularization, resulting in the ALICE [29] model. The recently proposed method [4] shares the same spirit. They lack an energy-based model on observed data.

1.3. Our Contributions

Our proposed formulation, which we call the *divergence triangle*, re-interprets and integrates the following elements in unsupervised generative learning: (1) maximum likelihood learning, (2) variational learning, (3) adversarial learning, (4) contrastive divergence, (5) wake-sleep algorithm. The learning is seamlessly integrated into a probabilistic framework based on KL divergence.

2. Learning Deep Probabilistic Models

In this section, we shall review the two probabilistic models, namely the generator model and the energy-based model, both of which are parametrized by convolutional neural networks [27, 26]. Then, we shall present the maximum likelihood learning algorithms for training these two models, respectively. Our presentation of the two maximum likelihood learning algorithms is unconventional. We seek to derive both algorithms based on the Kullback-Leibler divergence using the same scheme. This will set the stage for the divergence triangle.

2.1. Generator Model and Energy-based Model

The generator model [12, 37, 25, 38, 33] is a generalization of the factor analysis model [39],

$$z \sim \mathrm{N}(0, I_d), \; x = g_\theta(z) + \epsilon, \tag{1}$$

where g_θ is a top-down mapping parametrized by a deep network with parameters θ. It maps the d-dimensional latent vector z to the D-dimensional signal x. $\epsilon \sim \mathrm{N}(0, \sigma^2 I_D)$ and is independent of z. In general, the model is defined by the prior distribution $p(z)$ and the conditional distribution $p_\theta(x|z)$. The complete-data model $p_\theta(z, x) = p(z)p_\theta(x|z)$. The observed-data model is $p_\theta(x) = \int p_\theta(z, x)dz$. The posterior distribution is $p_\theta(z|x) = p_\theta(z, x)/p_\theta(x)$. See the diagram (a) below.

Top-down mapping	Bottom-up mapping
hidden vector z	energy $-f_\alpha(x)$
$\Downarrow$	$\Uparrow$
signal $x \approx g_\theta(z)$	signal x
(a) Generator model	(b) Energy-based model

A complementary model is the energy-based model [35, 5, 31, 48], where $-f_\alpha(x)$ defines the energy of x, and a

low energy x is assigned a high probability. Specifically, we have the following probability model

$$\pi_\alpha(x) = \frac{1}{Z(\alpha)} \exp\left[f_\alpha(x)\right], \tag{2}$$

where $f_\alpha(x)$ is parametrized by a bottom-up deep network with parameters α, and $Z(\alpha)$ is the normalizing constant. If $f_\alpha(x)$ is linear in α, the model becomes the familiar exponential family model in statistics or the Gibbs distribution in statistical physics. We may consider π_α an evaluator, where f_α assigns the value to x, and π_α evaluates x by a normalized probability distribution. See the diagram (b) above.

The energy-based model π_α defines explicit log-likelihood via $f_\alpha(x)$, even though $Z(\alpha)$ is intractable. However, it is difficult to sample from π_α. The generator model p_θ can generate x directly by first generating $z \sim p(z)$, and then transforming z to x by $g_\theta(z)$. But it does not define an explicit log-likelihood of x.

In the context of inverse reinforcement learning [54, 1] or inverse optimal control, x is action and $-f_\alpha(x)$ defines the cost function or $f_\alpha(x)$ defines the value function or the objective function.

2.2. Maximum Likelihood Learning

Let $q_{\text{data}}(x)$ be the true distribution that generates the training data. Both the generator p_θ and the energy-based model π_α can be learned by maximum likelihood. For large sample, the maximum likelihood amounts to minimizing the Kullback-Leibler divergence $\text{KL}(q_{\text{data}} \| p_\theta)$ over θ, and minimizing $\text{KL}(q_{\text{data}} \| \pi_\alpha)$ over α, respectively. The expectation $\text{E}_{q_{\text{data}}}$ can be approximated by sample average.

2.2.1 EM-type Learning of Generator Model

To learn the generator model p_θ, we seek to minimize $\text{KL}(q_{\text{data}}(x) \| p_\theta(x))$ over θ. Suppose in an iterative algorithm, the current θ is θ_t. We can fix θ_t at any place we want, and vary θ around θ_t.

We can write

$$\begin{aligned} &\text{KL}(q_{\text{data}}(x) p_{\theta_t}(z|x) \| p_\theta(z, x)) = \\ &\text{KL}(q_{\text{data}}(x) \| p_\theta(x)) + \text{KL}(p_{\theta_t}(z|x) \| p_\theta(z|x)). \end{aligned} \tag{3}$$

In the EM algorithm [7], the left hand side is the surrogate objective function. This surrogate function is more tractable than the true objective function $\text{KL}(q_{\text{data}}(x) \| p_\theta(x))$ because $q_{\text{data}}(x) p_{\theta_t}(z|x)$ is a distribution of the complete data, and $p_\theta(z, x)$ is the complete-data model.

We can write (3) as

$$S(\theta) = K(\theta) + \tilde{K}(\theta). \tag{4}$$

The geometric picture is that the surrogate objective function $S(\theta)$ is above the true objective function $K(\theta)$, i.e., S majorizes (upper bounds) K, and they touch each other at θ_t, so that $S(\theta_t) = K(\theta_t)$ and $S'(\theta_t) = K'(\theta_t)$. The reason is that $\tilde{K}(\theta_t) = 0$ and $\tilde{K}'(\theta_t) = 0$. See Figure 1.

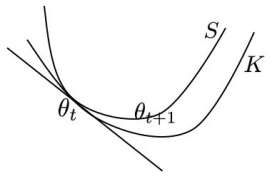

Figure 1. The surrogate S majorizes (upper bounds) K, and they touch each other at θ_t with the same tangent.

$q_{\text{data}}(x) p_{\theta_t}(z|x)$ gives us the complete data. Each step of EM fits the complete-data model $p_\theta(z, x)$ by minimizing the surrogate $S(\theta)$,

$$\theta_{t+1} = \arg\min_\theta \text{KL}(q_{\text{data}}(x) p_{\theta_t}(z|x) \| p_\theta(z, x)), \tag{5}$$

which amounts to maximizing the complete-data log-likelihood. By minimizing S, we will reduce $S(\theta)$ relative to θ_t, and we will reduce $K(\theta)$ even more, relative to θ_t, because of the majorization picture.

We can also use gradient descent to update θ. Because $S'(\theta_t) = K'(\theta_t)$, and we can place θ_t anywhere, we have

$$\begin{aligned} &-\frac{\partial}{\partial \theta} \text{KL}(q_{\text{data}}(x) \| p_\theta(x)) \\ &= \text{E}_{q_{\text{data}}(x) p_\theta(z|x)} \left[\frac{\partial}{\partial \theta} \log p_\theta(z, x) \right]. \end{aligned} \tag{6}$$

To implement the above updates, we need to compute the expectation with respect to the posterior distribution $p_\theta(z|x)$. It can be approximated by MCMC such as Langevin dynamics or HMC [34]. Both require gradient computations that can be efficiently accomplished by back-propagation. We have learned the generator using such learning method [13].

2.2.2 Self-critic Learning of Energy-based Model

To learn the energy-based model π_α, we seek to minimize $\text{KL}(q_{\text{data}}(x) \| \pi_\alpha(x))$ over α. Suppose in an iterative algorithm, the current α is α_t. We can fix α_t at any place we want, and vary α around α_t.

Consider the following contrastive divergence

$$\text{KL}(q_{\text{data}}(x) \| \pi_\alpha(x)) - \text{KL}(\pi_{\alpha_t}(x) \| \pi_\alpha(x)). \tag{7}$$

We can use the above as surrogate function, which is more tractable than the true objective function, since the $\log Z(\alpha)$ term is canceled out. Specifically, we can write (7) as

$$S(\alpha) = K(\alpha) - \tilde{K}(\alpha) \tag{8}$$

$$= -(\text{E}_{q_{\text{data}}}[f_\alpha(x)] - \text{E}_{\pi_{\alpha_t}}[f_\alpha(x)]) + \text{const} \tag{9}$$

The geometric picture is that the surrogate function $S(\alpha)$ is below the true objective function $K(\alpha)$, i.e., S minorizes (lower bounds) K, and they touch each other at α_t, so that $S(\alpha_t) = K(\alpha_t)$, and $S'(\alpha_t) = K'(\alpha_t)$. The reason is that $\tilde{K}(\alpha_t) = 0$ and $\tilde{K}'(\alpha_t) = 0$. See Figure 2.

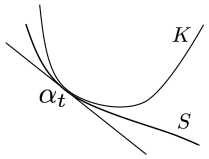

Figure 2. The surrogate S minorizes (lower bounds) K, and they touch each other at α_t with the same tangent.

Because S minorizes K, we do not have a EM-like update. However, we can still use gradient descent to update α, where the derivative is

$$K'(\alpha_t) = S'(\alpha_t) = -(\mathrm{E}_{q_{\mathrm{data}}}[f'_{\alpha_t}(x)] - \mathrm{E}_{\pi_{\alpha_t}}[f'_{\alpha_t}(x)]), \quad (10)$$

where

$$f'_{\alpha_t}(x) = \frac{\partial}{\partial \alpha} f_\alpha(x)\Big|_{\alpha_t}. \quad (11)$$

Since we can place α_t anywhere, we have

$$\begin{aligned} &-\frac{\partial}{\partial \alpha}\mathrm{KL}(q_{\mathrm{data}}(x)\|\pi_\alpha(x)) \\ &= \mathrm{E}_{q_{\mathrm{data}}}\left[\frac{\partial}{\partial \alpha} f_\alpha(x)\right] - \mathrm{E}_{\pi_\alpha}\left[\frac{\partial}{\partial \alpha} f_\alpha(x)\right]. \quad (12) \end{aligned}$$

To implement the above update, we need to compute the expectation with respect to the current model π_{α_t}. It can be approximated by MCMC such as Langevin dynamics or HMC that samples from π_{α_t}. It can be efficiently implemented by gradient computation via back-propagation. We have trained the energy-based model using such learning method [31, 48].

The above learning algorithm has an adversarial interpretation. Updating α_t to α_{t+1} by following the gradient of $S(\alpha) = \mathrm{KL}(q_{\mathrm{data}}(x)\|\pi_\alpha(x)) - \mathrm{KL}(\pi_{\alpha_t}(x)\|\pi_\alpha(x)) = -(\mathrm{E}_{q_{\mathrm{data}}}[f_\alpha(x)] - \mathrm{E}_{\pi_{\alpha_t}}[f_\alpha(x)]) + \mathrm{const}$, we seek to decrease the first KL-divergence, while we will increase the second KL-divergence, or we seek to shift the value function $f_\alpha(x)$ toward the observed data and away from the synthesized data generated from the current model. That is, the model π_α criticizes its current version π_{α_t}, i.e., the model is its own adversary or its own critic.

2.2.3 Similarity and Difference

In both models, at θ_t or α_t, we have $S = K$, $S' = K'$, because $\tilde{K} = 0$ and $\tilde{K}' = 0$.

The difference is that in the generator model, $S = K + \tilde{K}$, whereas in energy-based model, $S = K - \tilde{K}$.

In the generator model, if we replace the intractable $p_{\theta_t}(z|x)$ by the inference model $q_\phi(z|x)$, we get VAE.

In energy-based model, if we replace the intractable $\pi_{\alpha_t}(x)$ by the generator $p_\theta(x)$, we get adversarial contrastive divergence (ACD). The negative sign in front of $\tilde{K}$ is the root of the adversarial learning.

3. Divergence Triangle: Integrating Adversarial and Variational Learning

In this section, we shall first present the divergence triangle, emphasizing its compact symmetric and anti-symmetric form. Then, we shall show that it is an re-interpretation and integration of existing methods, in particular, VAE [25, 38, 33] and ACD [23, 6].

3.1. Loss Function

Suppose we observe training examples $\{x_{(i)} \sim q_{\mathrm{data}}(x)\}_{i=1}^n$ where $q_{\mathrm{data}}(x)$ is the unknown data distribution. $\pi_\alpha(x) \propto \exp[f_\alpha(x)]$ with energy function $-f_\alpha$ denotes the energy-based model with parameters α. The generator model $p(z)p_\theta(x|z)$ has parameters θ and latent vector z. It is trivial to sample the latent distribution $p(z)$ and the generative process is defined as $z \sim p(z)$, $x \sim p_\theta(x|z)$.

The maximum likelihood learning algorithms for both the generator and energy-based model require MCMC sampling. We modify the maximum likelihood KL-divergences by proposing a divergence triangle criterion, so that the two models can be learned jointly without MCMC. In addition to the generator p_θ and energy-based model π_α, we also include an inference model $q_\phi(z|x)$ in the learning scheme. Such an inference model is a key component in the variational auto-encoder [25, 38, 33]. The inference model $q_\phi(z|x)$ with parameters ϕ maps from the data space to latent space. In the context of EM, $q_\phi(z|x)$ can be considered an imputor that imputes the missing data z to get the complete data (z, x).

The three models above define joint distributions over z and x from different perspectives. The two marginals, i.e., empirical data distribution $q_{\mathrm{data}}(x)$ and latent prior distribution $p(z)$, are known to us. The goal is to harmonize the three joint distributions so that the competition and cooperation between different loss terms improves learning.

The divergence triangle involves the following three joint distributions on (z, x):

1. Q-distribution: $Q(z, x) = q_{\mathrm{data}}(x)q_\phi(z|x)$.
2. P-distribution: $P(z, x) = p(z)p_\theta(x|z)$.
3. Π-distribution: $\Pi(z, x) = \pi_\alpha(x)q_\phi(z|x)$.

We propose to learn the three models p_θ, π_α, q_ϕ by the

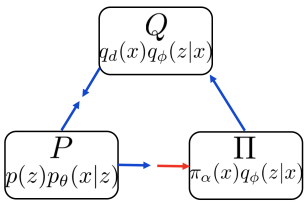

Figure 3. Divergence triangle is based on the Kullback-Leibler divergences between three joint distributions of (z, x). The blue arrow indicates the "running toward" behavior and the red arrow indicates the "running away" behavior.

following divergence triangle loss functional $\mathcal{D}$

$$\max_{\alpha} \min_{\theta} \min_{\phi} \mathcal{D}(\alpha, \theta, \phi),$$
$$\mathcal{D} = \mathrm{KL}(Q\|P) + \mathrm{KL}(P\|\Pi) - \mathrm{KL}(Q\|\Pi). \quad (13)$$

See Figure 3 for illustration. The divergence triangle is based on the three KL-divergences between the three joint distributions on (z, x). It has a symmetric and anti-symmetric form, where the anti-symmetry is due to the negative sign in front of the last KL-divergence and the maximization over α. The divergence triangle leads to the following dynamics between the three models: (1) Q and P seek to get close to each other. (2) P seeks to get close to Π. (3) π seeks to get close to q_{data}, but it seeks to get away from P, as indicated by the red arrow. Note that $\mathrm{KL}(Q\|\Pi) = \mathrm{KL}(q_{\text{data}}\|\pi_\alpha)$, because $q_\phi(z|x)$ is canceled out. The effect of (2) and (3) is that π gets close to q_{data}, while inducing P to get close to q_{data} as well, or in other words, P chases π_α toward q_{data}.

3.2. Unpacking the Loss Function

The divergence triangle integrates variational and adversarial learning methods, which are modifications of maximum likelihood.

3.2.1 Variational Learning

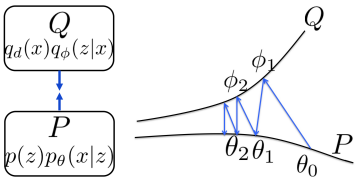

Figure 4. Variational auto-encoder (VAE) as joint minimization by alternating projection. Left: Interaction between the models. Right: Alternating projection. The two models run toward each other.

First, $\min_\theta \min_\phi \mathrm{KL}(Q\|P)$ captures the variational auto-encoder (VAE).

$$\begin{aligned} \mathrm{KL}(Q\|P) &= \mathrm{KL}(q_{\text{data}}(x)\|p_\theta(x)) \\ &+ \mathrm{KL}(q_\phi(z|x)\|p_\theta(z|x)), \end{aligned} \quad (14)$$

Recall $S = K + \tilde{K}$ in (4), if we replace the intractable $p_{\theta_t}(z|x)$ in (4) by the explicit $q_\phi(z|x)$, we get (14), so that we avoid MCMC for sampling $p_{\theta_t}(z|x)$.

We may interpret VAE as alternating projection between Q and P. See Figure 4 for illustration. If $q_\phi(z|x) = p_\theta(z|x)$, the algorithm reduces to the EM algorithm. The wake-sleep algorithm [17] is similar to VAE, except that it updates ϕ by $\min_\phi \mathrm{KL}(P\|Q)$ instead of $\min_\phi \mathrm{KL}(Q\|P)$, so that the wake-sleep algorithm does not have a single objective function.

The VAE $\min_\theta \min_\phi \mathrm{KL}(Q\|P)$ defines a cooperative game, with the dynamics that q_ϕ and p_θ run toward each other.

3.2.2 Adversarial Learning

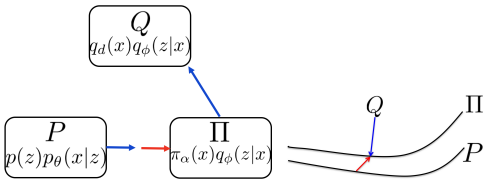

Figure 5. Adversarial contrastive divergence (ACD). Left: Interaction between the models. Red arrow indicates a chasing game, where the generator model chases the energy-based model, which runs toward the data distribution. Right: Contrastive divergence.

Next, consider the learning of the energy-based model model [23, 6]. Recall $S = K - \tilde{K}$ in (8), if we replace the intractable $\pi_{\alpha_t}(x)$ in (8) by $p_\theta(x)$, we get

$$\min_\alpha \max_\theta [\mathrm{KL}(q_{\text{data}}(x)\|\pi_\alpha(x)) - \mathrm{KL}(p_\theta(x)\|\pi_\alpha(x))], \quad (15)$$

or equivalently

$$\max_\alpha \min_\theta [\mathrm{KL}(p_\theta(x)\|\pi_\alpha(x)) - \mathrm{KL}(q_{\text{data}}(x)\|\pi_\alpha(x))], \quad (16)$$

so that we avoid MCMC for sampling $\pi_{\alpha_t}(x)$, and the gradient for updating α becomes

$$\frac{\partial}{\partial \alpha}[\mathrm{E}_{q_{\text{data}}}(f_\alpha(x)) - \mathrm{E}_{p_\theta}(f_\alpha(x))]. \quad (17)$$

Because of the negative sign in front of the second KL-divergence in (15), we need $\max_\theta$ in (15) or $\min_\theta$ in (16), so that the learning becomes adversarial. See Figure 5 for illustration. Inspired by [16], we call (15) the adversarial contrastive divergence (ACD). It underlies [23, 6].

The adversarial form (15) or (16) defines a chasing game with the following dynamics: the generator p_θ chases the energy-based model π_α in $\min_\theta \mathrm{KL}(p_\theta\|\pi_\alpha)$, the energy-based model π_α seeks to get closer to q_{data} and get away from p_θ. The red arrow in Figure 5 illustrates this chasing game. The result is that π_α lures p_θ toward q_{data}.

Figure 6. Generated samples. Left: generated samples on CIFAR-10 dataset. Right: generated samples on CelebA dataset.

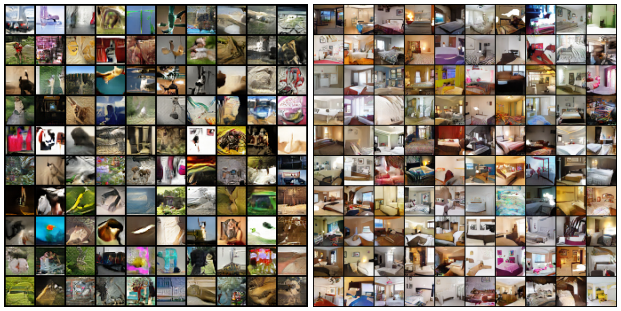

Figure 7. Generated samples. Left: 32×32 ImageNet. Right: 64×64 LSUN (bedroom).

In the idealized case, p_θ always catches up with π_α, then π_α will converge to the maximum likelihood estimate $\min_\alpha \mathrm{KL}(q_{\mathrm{data}} \| \pi_\alpha)$, and p_θ converges to π_α.

The updating of α by (17) bears similarity to Wasserstein GAN (WGAN) [2], but unlike WGAN, f_α defines a probability distribution π_α, and the learning of θ is based on $\min_\theta \mathrm{KL}(p_\theta(x) \| \pi_\alpha(x))$, which is a variational approximation to π_α. This variational approximation only requires knowing $f_\alpha(x)$, without knowing $Z(\alpha)$. However, unlike $q_\phi(z|x)$, $p_\theta(x)$ is still intractable, in particular, its entropy does not have a closed form. Thus, we can again use variational approximation, by changing the problem to $\min_\theta \min_\phi \mathrm{KL}(p(z)p_\theta(x|z) \| \pi_\alpha(x) q_\phi(z|x))$, i.e., $\min_\theta \min_\phi \mathrm{KL}(P \| \Pi)$, which is analytically tractable and which underlies [6]. In fact,

$$\mathrm{KL}(P \| \Pi) = \mathrm{KL}(p_\theta(x) \| \pi_\alpha(x)) + \mathrm{KL}(p_\theta(z|x) \| q_\phi(z|x)). \quad (18)$$

Thus, we can modify (16) into $\max_\alpha \min_\theta \min_\phi [\mathrm{KL}(P \| \Pi) - \mathrm{KL}(Q \| \Pi)]$, because again $\mathrm{KL}(Q \| \Pi) = \mathrm{KL}(q_{\mathrm{data}} \| \pi_\alpha)$.

Fitting the above together, we have the divergence triangle (13), which has a compact symmetric and anti-symmetric form.

3.3. Training Algorithm

The three models are parameterized by convolutional neural networks. Algorithm 1 outlines joint learning under the divergence triangle. In practice we use stochastic gradient descent and the expectations are replaced by the sample averages.

Algorithm 1 Joint Training for Divergence Triangle Model

Require:
training images $\{x_{(i)}\}_{i=1}^n$,
number of learning iterations T,
α, θ, $\phi \leftarrow$ initialized network parameters.

Ensure:
estimated parameters $\{\alpha, \theta, \phi\}$,
generated samples $\{\tilde{x}_{(i)}\}_{i=1}^{\tilde{n}}$.

Let $t \leftarrow 0$.
repeat
  $\{z_{(i)} \sim p(z)\}_{i=1}^{\tilde{M}}$.
  $\{\tilde{x}_{(i)} \sim p_\theta(x|z_{(i)})\}_{i=1}^{\tilde{M}}$.
  $\{x_{(i)} \sim q_{\mathrm{data}}(x)\}_{i=1}^{M}$.
  $\{\tilde{z}_{(i)} \sim q_\phi(z|x_{(i)})\}_{i=1}^{M}$.
  α**-step**: Given $\{\tilde{x}_{(i)}\}_{i=1}^{\tilde{M}}$ and $\{x_{(i)}\}_{i=1}^{M}$, update $\alpha \leftarrow \alpha + \eta_\alpha \frac{\partial}{\partial \alpha} \mathcal{D}$ with learning rate η_α.
  ϕ**-step**: Given $\{(z_{(i)}, \tilde{x}_{(i)})\}_{i=1}^{\tilde{M}}$ and $\{(\tilde{z}_{(i)}, x_{(i)})\}_{i=1}^{M}$, update $\phi \leftarrow \phi - \eta_\phi \frac{\partial}{\partial \phi} \mathcal{D}$, with learning rate η_ϕ.
  θ**-step**: Given $\{(z_{(i)}, \tilde{x}_{(i)})\}_{i=1}^{\tilde{M}}$ and $\{(\tilde{z}_{(i)}, x_{(i)})\}_{i=1}^{M}$, update $\theta \leftarrow \theta - \eta_\theta \frac{\partial}{\partial \theta} \mathcal{D}$, with learning rate η_θ (optional: multiple-step update).
  Let $t \leftarrow t + 1$.
until $t = T$

4. Experiments

The images are resized and scaled to $[-1, 1]$. The network parameters are initialized with zero-mean Gaussian with standard deviation 0.02 and optimized using Adam [24]. Network weights are decayed with rate 0.0005, and batch normalization [20] is used. The code is available at https://github.com/enijkamp/triangle.

4.1. Image Generation

4.1.1 Object Generation

For object categories, we test our model on two commonly-used datasets of natural images: CIFAR-10 and CelebA [30]. For CelebA face dataset, we randomly select 9,000 images for training and another 1,000 images for testing in reconstruction task. The face images are resized to 64×64 and CIFAR-10 images remain 32×32. The qualitative results of generated samples for objects are shown in Figure 6. We further evaluate our model using quantitative

Figure 8. Generated samples with $1,024 \times 1,024$ resolution drawn from $g_\theta(z)$ with 512-dimensional latent vector for CelebA-HQ.

Figure 9. High-resolution synthesis from the generator model $g_\theta(z)$ with linear interpolation in latent space for CelebA-HQ.

Model	VAE [25]	DCGAN [37]	WGAN [2]	CoopNet [46]	CEGAN [6]	ALI [10]	ALICE [29]	Ours
CIFAR-10 (IS)	4.08	6.16	5.76	6.55	7.07	5.93	6.02	**7.23**
CelebA (FID)	99.09	38.39	36.36	56.57	41.89	60.29	46.14	**31.92**

Table 1. Sample quality evaluation. Row 1: inception scores for CIFAR-10. Row 2: FID scores for CelebA.

Model	WS [17]	VAE [25]	ALI [10]	ALICE [29]	Ours
CIFAR-10	0.058	0.037	0.311	0.034	**0.028**
CelebA	0.152	0.039	0.519	0.046	**0.030**

Table 2. Test reconstruction evaluation. Row 1: MSE for CIFAR-10 test set. Row 2: MSE for 1,000 hold out set from CelebA.

evaluations which are based on the Inception Score (IS) [41] for CIFAR-10 and Frechet Inception Distance (FID) [32] for CelebA faces. We generate 50,000 random samples for the computation of the inception score and 10,000 random samples for the computation of the FID score. Table 1 shows the IS and FID scores of our model compared with VAE [25], DCGAN [37], WGAN [2], CoopNet [47], CEGAN [6], ALI [10], ALICE [29].

For the Inception Score on CIFAR-10, we borrowed the scores from relevant papers, and for FID score on 9,000 CelebA faces, we re-implemented or used available code with network structures similar to our model. The divergence triangle achieves competitive performance compared to recent baseline models.

4.1.2 Large-scale Dataset

We also train our model on large scale datasets including down-sampled 32×32 version of ImageNet [36, 40] (roughly 1 million images) and Large-scale Scene Understand (LSUN) dataset [49]. For the LSUN dataset, we consider the *bedroom*, *tower* and *Church ourdoor* categories which contains roughly 3 million, 0.7 million and 0.1 million images and were re-sized to 64×64. The network structures are similar with the ones used in object generation with twice the number of channels and batch normalization is used in all three models. Generated samples are shown on Figure 7.

4.1.3 High-resolution Synthesis

In this section, we recruit a layer-wise training scheme to learn models on CelebA-HQ [22] with resolutions of up to $1,024 \times 1,024$ pixels. Layer-wise training dates back to initializing deep neural networks by Restricted Boltzmann Machines to overcome optimization hurdles [18, 3]. The technique has been resurrected in progressive GANs [22], albeit the order of layer transitions is reversed such that top layers are trained first. This resembles a Laplacian Pyramid [8] in which images are generated in a coarse-to-fine fashion.

As in [22], the training starts with down-sampled images with a spatial resolution of 4×4 while progressively increasing the size of the images and number of layers. All three models are grown in synchrony where 1×1 convolutions project between RGB and feature. In contrast to [22], we do not require mini-batch discrimination to increase variation of $g_\theta(\cdot)$ nor gradient penalty to preserve 1-Lipschitz continuity of $f_\alpha(\cdot)$.

Figure 8 depicts high-fidelity synthesis in a resolution of $1,024 \times 1,024$ pixels sampled from the generator model $g_\theta(z)$ on CelebA-HQ. Figure 9 illustrates linear interpolation in latent space.

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

4.2. Test Image Reconstruction

In this experiment, we evaluate the reconstruction ability of our model for a hold-out testing image dataset. This is a strong indicator for the accuracy of our inference model. Specifically, if our divergence triangle model $\mathcal{D}$ is well-learned, then the inference model should match the true posterior of generator model, i.e., $q_\phi(z|x) \approx p_\theta(z|x)$. Therefore, given test signal x_{te}, its reconstruction $\tilde{x}_{te}$ should be close to x_{te}, i.e., $x_{te} \xrightarrow{q_\phi} z_{te} \xrightarrow{p_\theta} \tilde{x}_{te} \approx x_{te}$. Figure 10 shows the testing images and their reconstructions on CIFAR-10 and CelebA.

For CIFAR-10 we use the 10,000 pre-defined test images, while for CelebA we use 1,000 hold-out images that are unseen in training. The reconstruction quality is quantitatively measured by per-pixel mean square error (MSE). Table 2 shows the per-pixel MSE of our model compared to WS [17], VAE [25], ALI [10], ALICE [29].

4.3. Energy Landscape Mapping

In the following experiment, we evaluate the learned energy-based model by mapping the macroscopic structure of the energy landscape. A well-formed energy function partitions the image space into meaningful Hopfield basins of attraction [19]. In order to learn such energy-function, in Algorithm 1, we perform multiple θ-steps such that the samples $\{\tilde{x}_i\}_{i=1}^{\tilde{M}}$ are sufficiently "close" to the local minima of $-f_\alpha(x)$. Following [14], we map the structure of the energy function $-f_\alpha$. First, we identify energy minima. Then, we sort the minima from lowest energy to highest energy and sequentially group images if the energy barrier between two minima satisfies some threshold. This process is continued until all minima have been clustered. Figure 11 depicts a mapping of $-f_\alpha$ in the form of a disconnectivity-graph [44] and suggests that the learned energy function not only encodes meaningful images as minima, but also forms meaningful macroscopic structure.

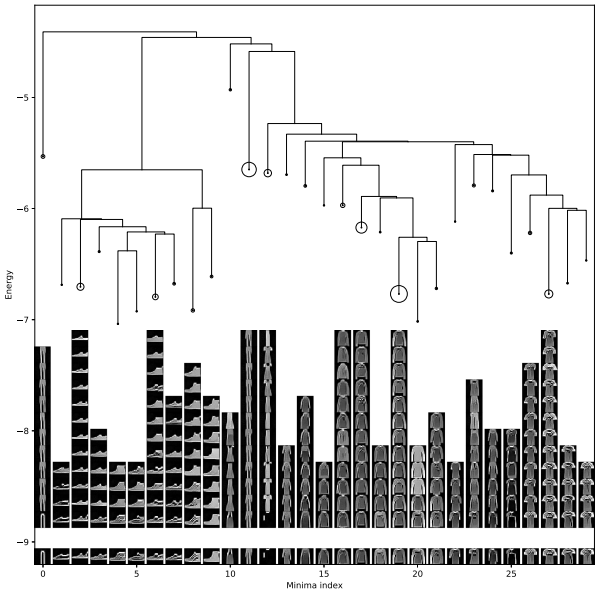

Figure 11. Disconnectivity-graph depicting the basin structure of the energy function for Fashion-MNIST. Each column represents basins members ordered by energy. Circle size indicates the total number of basin members. Vertical lines encode minima depth in terms of energy and horizontal lines depict the lowest known barrier at which two basins merge.

5. Conclusion

We propose a novel probabilistic framework, namely the divergence triangle, for joint learning of the energy-based model, the generator model, and the inference model. The divergence triangle forms a compact learning functional for three models and naturally unifies aspects of maximum likelihood estimation [13, 47], variational auto-encoder [25, 38, 33], adversarial learning [23, 6], contrastive divergence [15], and the wake-sleep algorithm [17].

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. We thank Dr. Tianfu Wu, Shuai Zhu and Bo Pang for helpful discussions.

References

[1] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1, 2004. 3

[2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223, 2017. 2, 6, 7

[3] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In *Advances in neural information processing systems*, pages 153–160, 2007. 7

[4] L. Chen, S. Dai, Y. Pu, E. Zhou, C. Li, Q. Su, C. Chen, and L. Carin. Symmetric variational autoencoder and connections to adversarial learning. In *International Conference on Artificial Intelligence and Statistics*, pages 661–669, 2018. 2

[5] J. Dai, Y. Lu, and Y.-N. Wu. Generative modeling of convolutional neural networks. *arXiv preprint arXiv:1412.6296*, 2014. 2

[6] Z. Dai, A. Almahairi, P. Bachman, E. Hovy, and A. Courville. Calibrating energy-based generative adversarial networks. *arXiv preprint arXiv:1702.01691*, 2017. 2, 4, 5, 6, 7, 8

[7] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977. 3

[8] E. L. Denton, S. Chintala, R. Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *NIPS*, pages 1486–1494, 2015. 7

[9] J. Donahue, P. Krähenbühl, and T. Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016. 2

[10] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016. 2, 7, 8

[11] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741, 1984. 1

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[13] T. Han, Y. Lu, S.-C. Zhu, and Y. N. Wu. Alternating back-propagation for generator network. In *AAAI*, volume 3, page 13, 2017. 1, 3, 8

[14] M. Hill, E. Nijkamp, and S.-C. Zhu. Building a telescope to look into high-dimensional image spaces. *arXiv preprint arXiv:1803.01043*, 2018. 8

[15] G. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, pages 1771–1800, 2002. 2, 8

[16] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002. 5

[17] G. E. Hinton, P. Dayan, B. J. Frey, and R. M. Neal. The" wake-sleep" algorithm for unsupervised neural networks. *Science*, 268(5214):1158–1161, 1995. 2, 5, 7, 8

[18] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006. 7

[19] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982. 8

[20] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 6

[21] L. Jin, J. Lazarow, and Z. Tu. Introspective learning for discriminative classification. In *Advances in Neural Information Processing Systems*, 2017. 2

[22] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 7

[23] T. Kim and Y. Bengio. Deep directed generative models with energy-based probability estimation. *arXiv preprint arXiv:1606.03439*, 2016. 2, 4, 5, 8

[24] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[25] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2, 4, 7, 8

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 2

[27] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 2

[28] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006. 2

[29] C. Li, H. Liu, C. Chen, Y. Pu, L. Chen, R. Henao, and L. Carin. Alice: Towards understanding adversarial learning for joint distribution matching. In *Advances in Neural Information Processing Systems*, pages 5495–5503, 2017. 2, 7, 8

[30] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015. 6

[31] Y. Lu, S.-C. Zhu, and Y. N. Wu. Learning FRAME models using CNN filters. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016. 1, 2, 4

[32] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet. Are gans created equal? a large-scale study. *arXiv preprint arXiv:1711.10337*, 2017. 7

[33] A. Mnih and K. Gregor. Neural variational inference and learning in belief networks. In *International Conference on Machine Learning*, pages 1791–1799, 2014. 2, 4, 8

[34] R. M. Neal. Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2, 2011. 1, 3

[35] J. Ngiam, Z. Chen, P. W. Koh, and A. Y. Ng. Learning deep energy models. In *International Conference on Machine Learning*, pages 1105–1112, 2011. 2

[36] A. v. d. Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016. 7

[37] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 2, 7

[38] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pages 1278–1286, 2014. 2, 4, 8

[39] D. B. Rubin and D. T. Thayer. Em algorithms for ml factor analysis. *Psychometrika*, 47(1):69–76, 1982. 2

[40] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 7

[41] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016. 7

[42] T. Tieleman. Training restricted boltzmann machines using approximations to the likelihood gradient. *ICML*, pages 1064–1071, 2008. 2

[43] Z. Tu. Learning generative models via discriminative approaches. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. 2

[44] D. J. Wales, M. A. Miller, and T. R. Walsh. Archetypal energy landscapes. *Nature*, 394(6695):758, 1998. 8

[45] Y. N. Wu, S. C. Zhu, and X. Liu. Equivalence of julesz ensembles and frame models. *International Journal of Computer Vision*, 38(3):247–265, 2000. 2

[46] J. Xie, Y. Lu, R. Gao, S.-C. Zhu, and Y. N. Wu. Cooperative training of descriptor and generator networks. *arXiv preprint arXiv:1609.09408*, 2016. 7

[47] J. Xie, Y. Lu, R. Gao, S.-C. Zhu, and Y. N. Wu. Cooperative training of descriptor and generator networks. *IEEE transactions on pattern analysis and machine intelligence (PAMI)*, 2018. 2, 7, 8

[48] J. Xie, Y. Lu, S.-C. Zhu, and Y. N. Wu. A theory of generative convnet. In *International Conference on Machine Learning*, pages 2635–2644, 2016. 1, 2, 4

[49] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 7

[50] J. Zhao, M. Mathieu, and Y. LeCun. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016. 2

[51] S.-C. Zhu. Statistical modeling and conceptualization of visual patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):691–712, 2003. 2

[52] S.-C. Zhu and D. Mumford. Grade: Gibbs reaction and diffusion equations. In *International Conference on Computer Vision*, pages 847–854, 1998. 2

[53] S.-C. Zhu, Y. N. Wu, and D. Mumford. Minimax entropy principle and its application to texture modeling. *Neural Computation*, 9(8):1627–1660, 1997. 2

[54] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey. Maximum entropy inverse reinforcement learning. 2008. 3