Learning Energy-Based Model
with Variational Auto-Encoder as Amortized Sampler

Jianwen Xie, Zilong Zheng, Ping Li

Cognitive Computing Lab
Baidu Research
10900 NE 8th St. Bellevue, WA 98004, USA
{jianwen.kenny, zlzheng.cs, pingli98}@gmail.com

Abstract

Due to the intractable partition function, training energy-based models (EBMs) by maximum likelihood requires Markov chain Monte Carlo (MCMC) sampling to approximate the gradient of the Kullback-Leibler divergence between data and model distributions. However, it is non-trivial to sample from an EBM because of the difficulty of mixing between modes. In this paper, we propose to learn a variational auto-encoder (VAE) to initialize the finite-step MCMC, such as Langevin dynamics that is derived from the energy function, for efficient amortized sampling of the EBM. With these amortized MCMC samples, the EBM can be trained by maximum likelihood, which follows an "analysis by synthesis" scheme; while the VAE learns from these MCMC samples via variational Bayes. We call this joint training algorithm the variational MCMC teaching, in which the VAE chases the EBM toward data distribution. We interpret the learning algorithm as a dynamic alternating projection in the context of information geometry. Our proposed models can generate samples comparable to GANs and EBMs. Additionally, we demonstrate that our model can learn effective probabilistic distribution toward supervised conditional learning tasks.

1 Introduction

Generative modeling of high-dimensional data is a very challenging and fundamental problem in both computer vision and machine learning communities. Energy-based generative model (Zhu, Wu, and Mumford 1998; LeCun et al. 2006) with the energy function parameterized by a deep neural network was first proposed by Xie et al. (2016), and has been drawing attention in the recent literature (Xie, Zhu, and Wu 2017; Gao et al. 2018; Xie et al. 2018c; Xie, Zhu, and Wu 2019; Du and Mordatch 2019; Nijkamp et al. 2019; Grathwohl et al. 2019), not only for its empirically powerful ability to learn highly complex probability distribution, but also for its theoretically fascinating aspects of representing high-dimensional data. Successful applications with energy-based generative frameworks have been witnessed in the field of computer vision, for example, video synthesis (Xie, Zhu, and Wu 2017, 2019), 3D volumetric shape synthesis (Xie et al. 2018c, 2020), unordered point cloud synthesis (Xie et al. 2021a), supervised image-to-image translation (Xie et al. 2019), and unpaired cross-domain visual

translation (Xie et al. 2021b). Other applications can be seen in natural language processing (Bakhtin et al. 2021), biology (Ingraham et al. 2018; Du et al. 2019), and inverse optimal control (Xu et al. 2019).

Energy-based generative models directly define an unnormalized probability density that is an exponential of the negative energy function, where the energy function maps the input variable to an energy scalar. Training an energy-based model (EBM) from observed data corresponds to finding an energy function, where observed data are assigned lower energies than unobserved ones. Synthesizing new data from the energy-based probability density can be achieved by a gradient-based Markov chain Monte Carlo (MCMC) method, which is an implicit and iterative generation process, to find low energy regions of the learned energy landscape; we refer readers to two excellent textbooks (Liu 2008; Barbu and Zhu 2020) and numerous references therein. Energy-based generative models, therefore, unify the generation and learning processes in a single model.

A persisting challenge in training an EBM of high-dimensional data via maximum likelihood estimation (MLE) is the calculation of the normalizing constant or the partition function, which requires a computationally intractable integral. Therefore, an MCMC sampling procedure, such as the Langevin dynamics or Hamiltonian Monte Carlo (Neal 2011), from the EBMs is typically used to approximate the gradient of the partition function during the model training. However, the MCMC is computationally expensive or even impractical, especially if the target distribution has multiple modes separated by highly low probability regions. In such a case, traversing modes becomes very difficult and unlikely because different MCMC chains easily get trapped by different local modes.

To tackle the above challenge, with the inspiration of the idea of amortized generation in Kim and Bengio (2016); Xie et al. (2018b), we propose to train a directed latent variable model as an approximate sampler that generates samples by deterministic transformation of independent and identically distributed random samples drawn from Gaussian distribution. Such an ancestral sampler can efficiently provide a good initial point of the iterative MCMC sampling of the EBM to avoid a long computation time to generate convergent samples. We call this process of first running an ancestral sampling by a latent variable model and then re-

vising the samples by a finite-step Langevin dynamics derived from an EBM the *Ancestral Langevin Sampling* (ALS). ALS takes advantages of both Langevin sampling and ancestral sampling. First, because the ancestral sampler connects the low-dimensional Gaussian distribution with the high-dimensional data distribution, traversing modes of the data distribution becomes more tractable and practical by sampling from the low-dimensional latent space. Secondly, the Langevin sampler is an attractor-like dynamics that can refine the initial samples by attracting them to the local modes of the energy function, thus making the initially generated samples stabler and more likely configurations.

From the learning perspective, by comparing the difference between the observed examples and the ALS examples, the EBM can find its way to shift its density toward the data distribution via MLE. The ALS with a small number of Langevin steps can accelerate the training of the EBM in terms of convergence speed. To approximate the Langevin sampler and serves as a good MCMC initializer, the latent variable model learns from the evolving EBM by treating the ALS examples at each iteration as training data. Different from Kim and Bengio (2016); Xie et al. (2018b), we follow the variational Bayes (Kingma and Welling 2014) to train the latent variable model by recruiting an approximate but computationally efficient inference model, which is typically an encoder network. Specifically, after the EBM revises the initial examples provided by the latent variable model, the inference model infers the latent variables of the revised examples, and then the latent variable model updates its mapping function by regressing the revised examples on their corresponding inferred latent codes. The inference model and the latent variable model form a modified variational auto-encoder (VAE) (Kingma and Welling 2014) that learns from evolving ALS samples, which are MCMC samples from the EBMs. In this framework, the EBM provides infinite batches of fresh MCMC examples as training data to the VAE model. The learning of the VAE are affected by the EBM. While providing help to the EBM in sampling, the VAE learns to chase the EBM, which runs towards the data distribution with the efficient sampling, ALS. Within the VAE, the inference model and the posterior of the latent variable model get close to each other via maximizing the variational lower bound of the log likelihood of the ALS samples. In other words, the latent variable model is trained with both variational inference of the inference model and MCMC teaching of the EBM. We call this the *Variational MCMC teaching*.

Moreover, the generative framework can be easily generalized to the conditional model by involving a conditional EBM and a conditional VAE, for representing a distribution of structured output given another structured input. This conditional model is very useful and can be applied to plenty of computer vision tasks, such as image inpainting etc.

Concretely, our contributions can be summarized below:

1. We present a new framework to train energy-based models (EBMs), where a VAE is jointly trained via MCMC teaching to fast initialize the Langevin dynamics of the EBM for its maximum likelihood learning. The amortized sampler is called *ancestral Langevin sampler*.

2. Our model provides a new strategy that we call *variational MCMC teaching* to train latent variable model, where an EBM and an inference model are simultaneously trained to provide infinite training examples and efficient approximate inference for the latent variable model, respectively.

3. We naturally unify the maximum likelihood learning, variational inference, and MCMC teaching in a single framework to induce maximum likelihood learning of all the probability models.

4. We provide an information geometric understanding of the proposed joint training algorithm. It can be interpreted as a dynamic alternating projection.

5. We provide strong empirical results on unconditional image modeling and conditional predictive learning to corroborate the proposed method.

2 Related Work

There are three types of interactions inside our model. The inference model and the latent variable model are trained in a variational inference scheme (Kingma and Welling 2014), the energy-based model (EBM) and the latent variable model are trained in a cooperative learning scheme (Xie et al. 2018b), and also the EBM and the data distribution forms an MCMC-based maximum likelihood estimation or "analysis by synthesis" learning scheme (Xie et al. 2016; Du and Mordatch 2019; Nijkamp et al. 2019).

Energy-based density estimation. The maximum likelihood estimation of the energy-based model (Zhu, Wu, and Mumford 1998; Wu, Zhu, and Liu 2000; LeCun et al. 2006; Hinton 2012; Xie et al. 2014; Lu, Zhu, and Wu 2016; Xie et al. 2016), follows what Grenander and Miller (2007) call "analysis by synthesis" scheme, where, at each iteration, the computation of the gradient of the log-likelihood requires MCMC sampling, such as the Gibbs sampling (Geman and Geman 1984), or Langevin dynamics. To overcome the computational hurdle of MCMC, the contrastive divergence (Hinton 2002), which is an approximate maximum-likelihood, initializes the MCMC with training data in learning the EBM. The noise-contrastive estimation (Gutmann and Hyvärinen 2012) of the EBM turns a generative learning problem into a discriminative learning one by preforming nonlinear logistic regression to discriminate the observed examples from some artificially generated noise examples. Nijkamp et al. (2019) propose to learn EBM with non-convergent non-persistent short-run MCMC as a flow-based generator, which can be useful for synthesis and reconstruction. The training of the EBM in our framework still follows "analysis by synthesis", except that the synthesis is performed by the *ancestral Langevin sampling*.

Training an EBM jointly with a complementary model. To avoid MCMC sampling of the EBM, Kim and Bengio (2016) approximate it by a latent variable model trained by minimizing the Kullback-Leibler (KL) divergence from the latent variable model to the EBM. It involves

an intractable entropy term, which is problematic if it is ignored. The gap between the latent variable model and the EBM due to their imbalanced model design may still cause bias or model collapse in training. We bridge the gap by taking back the MCMC to serve as an attractor-like dynamics to refine any imperfection of the latent variable model in the learned VAE. Xie et al. (2018b); Song and Ou (2018) study a similar problem. In comparison with Xie et al. (2018b), which either uses another MCMC to compute the intractable posterior of the latent variable model or directly ignores the inference step for approximation in their experiments, our framework learns a tractable variational inference model for training the latent variable model. The proposed framework is a variant of cooperative networks in Xie et al. (2018b).

3 Preliminary

In this section, we present the backgrounds of energy-based models (EBMs) and variational auto-encoders (VAEs), which will serve as foundations of the proposed framework.

3.1 EBM and Analysis by Synthesis

Let $x \in \mathbb{R}^D$ be the high-dimensional random variable, such as an input image. An EBM (also called Markov random field, Gibbs distribution, or exponential family model), with an energy function $U_\theta(x)$ and a set of trainable parameters θ, learns to associate a scalar energy value to each configuration of the random variable, such that more plausible configurations (e.g., observed training images) are assigned lower energy values. Formally, an EBM is defined as a probability density with the following form:

$$p_\theta(x) = \frac{1}{Z(\theta)} \exp[-U_\theta(x)], \qquad (1)$$

where $Z(\theta) = \int \exp[-U_\theta(x)]dx$ is a normalizing constant or a partition function depending on θ, and is analytically intractable to calculate due to high dimensionality of x. Following the EBM introduced in Xie et al. (2016), we can parameterize $U_\theta(x)$ by a bottom-up ConvNet with trainable weights θ and scalar output.

Assume a training dataset $\mathcal{D} = \{x_i, i = 1, ..., n\}$ is given and each data point is sampled from an unknown distribution $p_{\text{data}}(x)$. In order to use the EBM $p_\theta(x)$ to estimate the data distribution $p_{\text{data}}(x)$, we can minimize the negative log-likelihood of the observed data $L(\theta, \mathcal{D}) = -\frac{1}{n}\sum_{i=1}^{n} \log p_\theta(x_i)$, or equivalently the KL-divergence between the two distributions $\text{KL}(p_{\text{data}}(x)||p_\theta(x))$ by gradient-based optimization methods. The gradient to update parameters θ is computed by the following formula

$$\frac{\partial}{\partial\theta}\text{KL}(p_{\text{data}}(x)||p_\theta(x))$$
$$=\text{E}_{x\sim p_{\text{data}}(x)}\left[\frac{\partial U_\theta(x)}{\partial\theta}\right] - \text{E}_{\tilde{x}\sim p_\theta(x)}\left[\frac{\partial U_\theta(\tilde{x})}{\partial\theta}\right]. \qquad (2)$$

The two expectations in Eq. (2) are approximated by averaging over the observed examples $\{x_i\}$ and the synthesized examples $\{\tilde{x}_i\}$ that are sampled from the model $p_\theta(x)$, respectively. This will lead to an analysis by synthesis algorithm that iterates a synthesis step for image sampling and an analysis step for parameter learning.

Drawing samples from EBMs typically requires Markov chain Monte Carlo (MCMC) methods. If the data distribution $p_{\text{data}}(x)$ is complex and multimodal, the MCMC sampling from the learned model is challenging because it may take a long time to mix between modes. Thus, the ability to generate efficient and fair examples from the model becomes the key to training successful EBMs. In this paper, we will study amortized sampling for efficient training of the EBMs.

3.2 Latent Variable Model and Variational Inference

Consider a directed latent variable model of the form

$$z \sim \mathcal{N}(0, I_d), x = g_\alpha(z) + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2 I_D), \qquad (3)$$

where $z \in \mathbb{R}^d$ is a d-dimensional vector of latent variables following a Gaussian distribution $\mathcal{N}(0, I_d)$, I_d is a d-dimensional identity matrix, g_α is a nonlinear mapping function that is parameterized by a top-down deep neural network with trainable parameters α, and $\epsilon \in \mathbb{R}^D$ is the residual noise that is independent of z.

The marginal distribution of the model in Eq. (3) is $q_\alpha(x) = \int q_\alpha(x|z)q(z)dz$, where the prior distribution $q(z) = \mathcal{N}(0, I_d)$ and the conditional distribution of x given z is $q_\alpha(x|z) = \mathcal{N}(g_\alpha(z), \sigma^2 I_D)$. The posterior distribution is $q_\alpha(z|x) = q_\alpha(z, x)/q_\alpha(x) = q_\alpha(x|z)q(z)/q_\alpha(x)$. Both posterior distribution $q_\alpha(z|x)$ and marginal distribution $q_\alpha(x)$ are analytically intractable. As in Han et al. (2017), the model can be learned by maximum likelihood estimation or equivalently minimizing the KL-divergence $\text{KL}(p_{\text{data}}(x)||q_\alpha(x))$, whose gradient is given by

$$\frac{\partial}{\partial\alpha}\text{KL}(p_{\text{data}}(x)||q_\alpha(x))$$
$$=\text{E}_{p_{\text{data}}(x)q_\alpha(z|x)}\left[-\frac{\partial}{\partial\alpha}\log q_\alpha(z, x)\right]. \qquad (4)$$

MCMC methods can be used to compute the gradient in Eq. (4). For each data point x_i sampled from the data distribution, we infer the corresponding latent variable z_i by drawing samples from $q_\alpha(z|x)$ via MCMC methods, then the expectation term can be approximated by averaging over the sampled pairs $\{x_i, z_i\}$. However, MCMC sampling of the posterior distribution may also take a long time to converge. To avoid MCMC sampling from $q_\alpha(z|x)$, VAE (Kingma and Welling 2014) approximates $q_\alpha(z|x)$ by a tractable inference network, for example, a multivariate Gaussian with a diagonal covariance structure $\pi_\beta(z|x) \sim \mathcal{N}(\mu_\beta(x), \text{diag}(v_\beta(x)))$, where both $\mu_\beta(x)$ and $v_\beta(x)$ are d-dimensional outputs of encoding bottom-up networks of data point x, with trainable parameters β. With this reparameterization trick, the objective of VAE becomes to find α and β to minimize

$$\text{KL}(p_{\text{data}}(x)\pi_\beta(z|x)||q_\alpha(z, x))$$
$$=\text{KL}(p_{\text{data}}(x)||q_\alpha(x)) + \text{KL}(\pi_\beta(z|x)||q_\alpha(z|x)), \qquad (5)$$

which is a modification of the maximum likelihood estimation objective. Minimizing the left-hand side in Eq. (5) will also lead to a minimization of the first KL-divergence on the

right-hand side, which is the maximum likelihood estimation objective in Eq. (4). In this paper, we will propose to learn a latent variable model in the context of VAE as amortized sampler to train the EBM.

4 Methodology

We study to learn an EBM via MLE with a VAE as amortized sampler. The amortized sampler is achieved by integrating the latent variable model (the generator network in VAE) and the short-run MCMC of the EBM. We propose to jointly train EBM and VAE via *variational MCMC teaching*.

4.1 Ancestral Langevin Sampling

To learn the energy-based generative model in Eq. (1) and compute the gradient in Eq. (2), we might bring in a directed latent variable model $q_\alpha(x)$ to serve as a fast non-iterative sampler to initialize the iterative MCMC sampler guided by the energy function U_θ, for the sake of efficient MCMC convergence and mode traversal of the EBM. In our paper, we call the resulting amortized sampler the *ancestral Langevin sampler*, which draws a sample by first (i) sampling an initial example $\hat{x}$ via ancestral sampling, and then (ii) revising $\hat{x}$ with a finite-step Langevin update, that is

$$
\begin{aligned}
&\text{(i) } \hat{x} = g_\alpha(\hat{z}),\ \hat{z} \sim \mathcal{N}(0, I_d), \\
&\text{(ii) } \tilde{x}_{t+1} = \tilde{x}_t - \frac{\delta^2}{2}\frac{\partial U_\theta(\tilde{x}_t)}{\partial \tilde{x}} + \delta \mathcal{N}(0, I_D),\ \tilde{x}_0 = \hat{x},
\end{aligned} \tag{6}
$$

where $\hat{x}$ is the initial example generated by ancestral sampling, $\tilde{x}$ is the example generated by the Langevin dynamics, t indexes the Langevin time step, and δ is the step size. The Langevin dynamics is equivalent to a stochastic gradient descent algorithm that seeks to find the minimum of the objective function defined by $U_\theta(x)$.

Generally, in the original "analysis by synthesis" algorithm, the Langevin dynamics shown in Eq. (6)(ii) is initialized with a noise distribution, such as Gaussian distribution, i.e., $\tilde{x}_0 \sim \mathcal{N}(0, I_D)$, and this usually takes a long time to converge and is also non-stable in practise because the gradient-based MCMC chains can get trapped in the local modes when exploring the model distribution.

As to the *ancestral Langevin sampling* in Eq. (6), intuitively, if the latent variable model in Eq. (6)(i) can memorize the majority of the modes in $p_\theta(x)$ by low dimensional codes $\hat{z}$, then we can easily traverse among modes of the model distribution by simply sampling from $p(\hat{z}) = \mathcal{N}(0, I_d)$, because $p(\hat{z})$ is much smoother than $p_\theta(x)$. The short-run Langevin dynamics initialized with the output $\hat{x}$ of the latent variable model emphasizes on refining the detail of $\hat{x}$ by further searching for a better mode $\tilde{x}$ around $\hat{x}$. Ideally, if $p_\theta(x)$ and $q_\alpha(x)$ fit the data distribution $p_{\text{data}}(x)$ perfectly, the example $\hat{x}$ produced by the ancestral sampling will be exactly on the modes of $U_\theta(x)$. In this case, the following Langevin revision will not change the $\hat{x}$, i.e., $\tilde{x} = \hat{x}$. Otherwise, the Langevin update will further improve $\hat{x}$.

4.2 Variational MCMC Teaching

With $\{\tilde{x}_i\}_{i=1}^{\tilde{n}} \sim p_\theta(x)$ via *ancestral Langevin sampling* in Eq. (6), we can compute the gradient in Eq. (2) by

$$
\begin{aligned}
&\frac{\partial}{\partial \theta}\text{KL}(p_{\text{data}}(x)\|p_\theta(x)) \\
&\approx \frac{1}{n}\sum_{i=1}^{n}\frac{\partial U_\theta(x_i)}{\partial \theta} - \frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}}\frac{\partial U_\theta(\tilde{x}_i)}{\partial \theta}
\end{aligned} \tag{7}
$$

and then update θ by Adam (Kingma and Ba 2015). Consider in this iterative algorithm, the current model parameter θ and α are θ_t and α_t respectively. We use $\mathcal{M}_{\theta_t}$ to denote the Markov transition kernel of a finite-step Langevin dynamics that samples from the current distribution $p_{\theta_t}(x)$. We also use $\mathcal{M}_{\theta_t}q_{\alpha_t}(x) = \int M_{\theta_t}(x', x)q_{\alpha_t}(x')dx'$ to denote the marginal distribution obtained by running $\mathcal{M}_{\theta_t}$ initialized from current $q_{\alpha_t}(x)$. The MCMC-based MLE training of θ seeks to minimize the following objective at each iteration

$$
\begin{aligned}
\theta_{t+1} = \arg\min_\theta [&\text{KL}(p_{\text{data}}(x)\|p_\theta(x)) \\
&- \text{KL}(\mathcal{M}_{\theta_t}q_{\alpha_t}(x)\|p_\theta(x))],
\end{aligned} \tag{8}
$$

which is considered as a modified contrastive divergence in Xie et al. (2018b,a). Meanwhile, $q_{\alpha_{t+1}}(x)$ is learned based on how the finite steps of Langevin $\mathcal{M}_{\theta_t}$ revises the initial example $\{\hat{x}_i\}$ generated by $q_{\alpha_t}(x)$ to mimic the Langevin sampling. This is the energy-based MCMC teaching (Xie et al. 2018b,a) of $q_\alpha(x)$.

Although $q_\alpha(x)$ initializes the Langevin sampling of $\{\tilde{x}_i\}$, the corresponding latent variables of $\{\tilde{x}_i\}$ are no longer $\{\hat{z}_i\}$. To retrieve the latent variables of $\{\tilde{x}_i\}$, we propose to infer $\tilde{z} \sim \pi_\beta(z|\tilde{x})$, which is an approximate tractable inference network, and then learn α from complete data $\{\tilde{z}_i, \tilde{x}_i\}_{i=1}^{\tilde{n}}$ to minimize $\sum_i \|\tilde{x}_i - g_\alpha(\tilde{z}_i)\|^2$ (or equivalently maximize $\sum_i \log q_\alpha(\tilde{z}_i, \tilde{x}_i)$). To ensure $\pi_\beta(z|\tilde{x})$ to be an effective inference network that mimics the computation of the true inference procedure $\tilde{z} \sim q_\alpha(z|\tilde{x})$, we simultaneously learn β by minimizing $\text{KL}(\pi_\beta(z|x)\|q_\alpha(z|x))$, i.e., the reparameterization trick of the variational inference of $q_\alpha(x)$.

The learning of $\pi_\beta(z|x)$ and $q_\alpha(x|z)$ forms a VAE that treats $\{\tilde{x}_i\}$ as training examples. Because $\{\tilde{x}_i\}$ are dependent on θ and vary during training, the objective function of the VAE is non-static. This is essentially different from the original VAE that has a fixed training data. Suppose we have $\{\tilde{x}_i\}_{i=1}^{\tilde{n}} \sim \mathcal{M}_{\theta_t}q_{\alpha_t}(x)$ at the current iteration t, the VAE objective in our framework is the minimization of variational lower bound of the negative log likelihood of $\{\tilde{x}_i\}_{i=1}^{\tilde{n}}$, i.e.,

$$
\begin{aligned}
L(\alpha, \beta) = \sum_{i=1}^{\tilde{n}}[&-\log q_\alpha(\tilde{x}_i) \\
&+ \gamma \text{KL}(\pi_\beta(z_i|\tilde{x}_i)\|q_\alpha(z_i|\tilde{x}_i))],
\end{aligned} \tag{9}
$$

where γ is a hyper-parameter that specifies the importance of the KL-divergence term. Since when $\tilde{n} \to \infty$, we have

$$
\min_\alpha \sum_{i=1}^{\tilde{n}}[-\log q_\alpha(\tilde{x}_i)] = \min_\alpha \text{KL}(\mathcal{M}_{\theta_t}q_{\alpha_t}(x)\|q_\alpha(x)),
$$

thus Eq. (9) is equivalent to minimizing

$$\begin{aligned}&\mathrm{KL}(\mathcal{M}_{\theta_t}q_{\alpha_t}(x)||q_\alpha(x)) + \mathrm{KL}(\pi_\beta(z|x)||q_\alpha(z|x))\\&=\mathrm{KL}(\mathcal{M}_{\theta_t}q_{\alpha_t}(x)\pi_\beta(z|x)||q_\alpha(x|z)q(z)).\end{aligned} \quad (10)$$

Unlike the objective function of the maximum likelihood estimation $\mathrm{KL}(\mathcal{M}_{\theta_t}q_{\alpha_t}(x)||q_\alpha(x))$, which involves intractable marginal distribution $q_\alpha(x)$, the variational objective function is the KL-divergence between the joint distributions, which is tractable because $\pi_\beta(z|x)$ parameterized by an encoder is tractable. In comparison with the original VAE objective in Eq. (5), our VAE objective in Eq. (10) replaces $p_{\mathrm{data}}(x)$ by $\mathcal{M}_{\theta_t}q_{\alpha_t}(x)$. At each iteration, minimizing the variational objective in Eq. (10) will eventually decrease $\mathrm{KL}(\mathcal{M}_{\theta_t}q_{\alpha_t}(x)||q_\alpha(x))$. Since $q_\alpha(x)$ is learned in the context of both MCMC teaching (Xie et al. 2018a) and variational inference (Kingma and Welling 2014). We call this the *variational MCMC teaching*. Algorithm 1 describes the proposed joint training algorithm of EBM and VAE.

Algorithm 1 Cooperative training of EBM and VAE via variational MCMC teaching

---

Input: (a) training images $\{x_i\}_{i=1}^n$, (b) number of Langevin steps l
Output: (a) model parameters $\{\theta, \alpha, \beta\}$, (b) initial samples $\{\hat{x}_i\}_{i=1}^{\tilde{n}}$, (c) Langevin samples $\{\tilde{x}_i\}_{i=1}^{\tilde{n}}$

1: Let $t \leftarrow 0$, randomly initialize θ, α, and β.
2: **repeat**
3:　　**ancestral Langevin sampling**: For $i = 1, ..., \tilde{n}$, sample $\hat{z}_i \sim \mathcal{N}(0, I_d)$, then generate $\hat{x}_i = g(\hat{z}_i)$, and run l steps of Langevin revision starting from $\hat{x}_i$ to obtain $\tilde{x}_i$, each step following Eq. (6)(ii).
4:　　**modified contrastive divergence**: Treat $\{\tilde{x}\}_i^{\tilde{n}}$ as MCMC examples from $p_\theta(x)$, and update θ by Adam with the gradient computed according to Eq. (7).
5:　　**variational MCMC teaching**: Treat $\{\tilde{x}\}_i^{\tilde{n}}$ as training data, update α and β by minimizing VAE objective in Eq. (9) via Adam.
6:　　Let $t \leftarrow t + 1$
7: **until** $t = T$

Figure 1 shows a comparison of the basic ideas of different types of MCMC teaching strategies. Figure 1(a) and (b) illustrate the diagrams of the original MCMC teaching and its fast variant in Xie et al. (2018a), respectively. Figure 1(c) displays the proposed *variational MCMC teaching* algorithm. Our framework in Figure 1(c) involves three models and adopts the reparameterization trick for inference, which is different from Figure 1(a) and (b).

4.3　Optimality of the Solution
In this section, we present a theoretical understanding of the framework presented in Section 4. A Nash equilibrium of the model is a triplet $(\hat{\theta}, \hat{\alpha}, \hat{\beta})$ that satisfies:

$$\hat{\theta} = \arg\min_\theta[\mathrm{KL}(p_{\mathrm{data}}(x)||p_\theta(x)) - \mathrm{KL}(\mathcal{M}_{\hat{\theta}}q_{\hat{\alpha}}(x)||p_\theta(x))], \quad (11)$$

$$\hat{\alpha} = \arg\min_\alpha[\mathrm{KL}(\mathcal{M}_{\hat{\theta}}q_{\hat{\alpha}}(x)||q_\alpha(x)) + \mathrm{KL}(\pi_{\hat{\beta}}(z|x)||q_\alpha(z|x))], \quad (12)$$

$$\hat{\beta} = \arg\min_\beta \mathrm{KL}(\pi_\beta(z|x)||q_{\hat{\alpha}}(z|x)). \quad (13)$$

We show that below if $(\hat{\theta}, \hat{\alpha}, \hat{\beta})$ is a Nash equilibrium of the model, then $p_{\hat{\theta}} = q_{\hat{\alpha}} = p_{\mathrm{data}}$.

In Eq. (12) and Eq. (13), the tractable encoder $\pi_{\hat{\beta}}(z|x)$ seeks to approximate the analytically intractable posterior distribution $q_{\hat{\alpha}}(z|x)$ via a joint minimization. When $\mathrm{KL}(\pi_{\hat{\beta}}(z|x)||q_{\hat{\alpha}}(z|x)) = 0$, then the second KL-divergence term in Eq. (12) vanishes, thus reducing Eq. (12) to $\min_\alpha \mathrm{KL}(\mathcal{M}_{\hat{\theta}}q_{\hat{\alpha}}(x)||q_\alpha(x))$, which means that $q_{\hat{\alpha}}$ seeks to be a stationary distribution of $\mathcal{M}_{\hat{\theta}}$, which is $p_{\hat{\theta}}$. Formally speaking, when $\min_\alpha \mathrm{KL}(\mathcal{M}_{\hat{\theta}}q_{\hat{\alpha}}(x)||q_\alpha(x)) = 0$, then $\mathcal{M}_{\hat{\theta}}q_{\hat{\alpha}}(x) = q_{\hat{\alpha}}(x)$, that is, $q_{\hat{\alpha}}$ converges to the stationary distribution $p_{\hat{\theta}}$, therefore we have $q_{\hat{\alpha}}(x) = p_{\hat{\theta}}(x)$. As a result, the second KL-divergence in Eq. (11) vanishes because $\mathrm{KL}(\mathcal{M}_{\hat{\theta}}q_{\hat{\alpha}}(x)||p_{\hat{\theta}}(x)) = \mathrm{KL}(\mathcal{M}_{\hat{\theta}}p_{\hat{\theta}}(x)||p_{\hat{\theta}}(x)) = 0$. Eq. (11) is eventually reduced to minimizing the first KL-divergence $\mathrm{KL}(p_{\mathrm{data}}(x)||p_{\hat{\theta}}(x))$, thus, $p_{\hat{\theta}}(x) = p_{\mathrm{data}}(x)$. The overall effect of the algorithm is that the EBM p_θ runs toward the data distribution p_{data} while inducing the latent variable model q_α to get close to the data distribution p_{data} as well, because q_α chases p_θ toward p_{data}, i.e., $q_\alpha \rightarrow p_\theta \rightarrow p_{\mathrm{data}}$, thus $q_{\hat{\alpha}} = p_{\hat{\theta}} = p_{\mathrm{data}}$. In other words, the joint training algorithm can lead to MLE of q_α and p_θ.

4.4　Conditional Predictive Learning
The proposed framework can be generalized to supervised learning of the conditional distribution of an output x given an input y, where both input and output are high-dimensional structured variables and may belong to two different modalities. We generalize the framework by turning both EBM and latent variable model into conditional ones. Specifically, the conditional EBM $p_\theta(x|y)$ represents a conditional distribution of x given y by using a joint energy function $U_\theta(x, y)$, the conditional latent variable model $q_\alpha(x|y, z)$ generates x by mapping y and a vector of latent Gaussian noise variables z together via $x = g_\alpha(y, z)$, and the conditional inference network $\pi_\beta(z|y, x) \sim \mathcal{N}(\mu_\beta(x, y), v_\beta(x, y))$, where $\mu_\beta(x, y)$ and
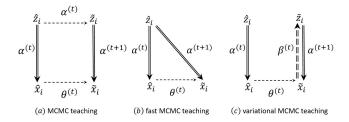

Figure 1: Diagrams of different types of MCMC teaching algorithms. (a) original MCMC teaching with an MCMC-based inference process. (b) fast MCMC teaching without an inference step. (c) *variational MCMC teaching*. The double-solid-line arrows indicate generation and reconstruction by the latent variable model with parameters α. The dashed-line arrows indicate Langevin dynamics guided by θ in the latent space or data space. The double-dashed-line arrow indicates inference and encoding by the inference model with β.

$v_\beta(x, y)$ are outputs of an encoder network taking x and y as inputs. $q_\alpha(x|y, z)$ and $\pi_\beta(z|x, y)$ form a conditional VAE (Sohn, Lee, and Yan 2015). Both the latent variable z in the latent variable model and the Langevin dynamics in the EBM allow for randomness in such a conditional mapping, thus making the proposed model suitable for representing one-to-many mapping. Once the conditional model is trained, we can generate samples $\{\tilde{x}_i\}$ conditioned on an input y by following the *ancestral Langevin sampling* process. To use the model on prediction tasks, we can perform a deterministic generation as prediction without sampling, i.e., the conditional latent variable model first generates an initial prediction via $z^* = \mathrm{E}(z), \hat{x}_i = g_\alpha(y_i, z^*)$, and then conditional EBM refines $\hat{x}$ by a finite steps of noise-disable Langevin dynamics $\tilde{x}_{t+1} = \tilde{x}_t - \frac{\delta^2}{2} \frac{\partial U(\tilde{x}_t, y_i)}{\partial \tilde{x}}$ with $\tilde{x}_{t=0} = \hat{x}$, which actually is a gradient descent that finds a local minimum around $\hat{x}$ in the learned energy function $U_\theta(x, y = y_i)$.

5 Information Geometric Understanding

In this section, we shall provide an information geometric understanding of the proposed learning algorithm, and show that our learning algorithm can be interpreted as a process of dynamic alternating projection within the framework of information geometry.

5.1 Three Families of Joint Distributions

The proposed framework includes three trainable models, i.e., energy-based model $p_\theta(x)$, inference model $\pi_\beta(z|x)$, and latent variable model $q_\alpha(x|z)$. They, along with the empirical data distribution $p_{\mathrm{data}}(x)$ and the Gaussian prior distribution $q(z)$, define three families of joint distributions over the latent variables z and the data x. Let us define

- Π-distribution: $\Pi(z, x) = p_{\mathrm{data}}(x)\pi_\beta(z|x)$
- Q-distribution: $Q(z, x) = q(z)q_\alpha(x|z)$
- P-distribution: $P(z, x) = p_\theta(x)\pi_\beta(z|x)$

In the context of information geometry, the above three families of distributions can be represented by three different manifolds. Each point of the manifold stands for a probability distribution with a certain parameter.

The *variational MCMC teaching* that we proposed in this paper to train both EBM and VAE actually integrates variational learning and energy-based learning, which is a modification of maximum likelihood estimation. The training process alternates these two learning processes, and eventually leads to maximum likelihood solutions of all the models. We first try to understand each part separately below, and then we integrate them together to give a final interpretation.

5.2 Variational Learning as Alternating Projection

The original variational learning algorithm, such as VAEs (Kingma and Welling 2014), is to learn $\{\alpha, \beta\}$ from training data $p_{\mathrm{data}}(x)$, whose objective function is a joint minimization $\min_\beta \min_\alpha \mathrm{KL}(\Pi||Q)$. However, in our learning algorithm, the VAE component learns to mimic the EBM at each iteration by learning from its generated examples. Thus, given θ_t at iteration t, our VAE objective becomes
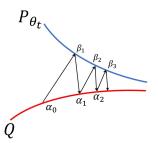

Figure 2: Variational learning is interpreted as an process of alternating projection between manifolds P_{θ_t} and Q. Manifold P_{θ_t} is represented by a blue curve and manifold Q is represented by a red curve. Each point of the red curve corresponds to a certain α, while each point of the blue curve corresponds to a certain β.

$\min_\beta \min_\alpha \mathrm{KL}(P_{\theta_t}||Q)$, where we put subscript θ_t in P to indicate that the P-distribution is associated with a fixed θ_t. The following reveals that $\mathrm{KL}(P_{\theta_t}||Q)$ is exactly the VAE loss we use in Eq. (10).

$$
\begin{aligned}
&\mathrm{KL}(P_{\theta_t}||Q) \\
=&\mathrm{KL}(p_{\theta_t}(x)\pi_\beta(z|x)||q_\alpha(x|z)q(z)) \\
=&\mathrm{KL}(p_{\theta_t}(x)||q_\alpha(x)) + \mathrm{KL}(\pi_\beta(z|x)||q_\alpha(z|x)) \\
=&\mathrm{KL}(\mathcal{M}_{\theta_t}q_{\alpha_t}(x)||q_\alpha(x)) + \mathrm{KL}(\pi_\beta(z|x)||q_\alpha(z|x)).
\end{aligned}
\tag{14}
$$

Minimizing the KL-divergence between two probability distributions can be interpreted as a projection from a probability distribution to a manifold (Cover 1999). Therefore, as illustrated in Figure 2, each manifold is visualized as a curve and the joint minimization in VAE in Eq. (14) can be interpreted as alternating projection (Han et al. 2019) between manifolds P_{θ_t} and Q, where π_β and q_α run toward each other and eventually converge at the intersection between manifolds P_{θ_t} and Q.

5.3 Energy-Based Learning as Manifold Shifting

With the examples generated by the *ancestral Langevin sampler*, the objective function of training the EBM is $\min_\theta \mathrm{KL}(\Pi||P)$, i.e., $\min_\theta \mathrm{KL}(p_{\mathrm{data}}||p_\theta)$. As illustrated in Figure 3, P_{θ_0} runs toward Π and seeks to match it. Each point in each curve represents a different β.
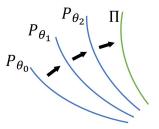

Figure 3: Energy-based learning is interpreted as a manifold shifting process from P_{θ_0} to Π, where θ_0 denotes the initial θ at time 0. Manifolds $\{P_{\theta_t}\}$ are represented by blue curves, while manifold Π is represented by a green curve.

5.4 Integrating Energy-Based Learning and Variational Learning as Dynamic Alternating Projection

The joint training of p_θ, π_β, q_α in the proposed framework integrates energy-based learning and variational learning, which can be interpreted as a dynamic alternating projection between Q and P, where Q is static but P is changeable and keeps shifting toward Π. See Figure 4 for an illustration. Ideally, P matches Π, i.e., $P_{\hat\theta} = \Pi$. The alternating projection would converge at the intersection point among Q, P and Π (see Figure 5), where we have $\min_\alpha \min_\beta \mathrm{KL}(\Pi||Q)$, which is the objective of the original VAE. In other words, Q and P get close to each other, while P seeks to get close to Π. In the end, q_α chases p_θ towards p_{data}.
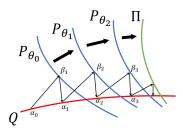

Figure 4: *Variational MCMC teaching* as dynamic alternating projection. Manifolds P, Q, and Π are represented by blue, red, and green curves, respectively.
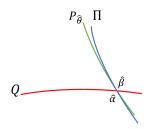

Figure 5: Convergent point of the dynamic alternating projection. Triplet $(\hat\theta, \hat\alpha, \hat\beta)$ is the Nash equilibrium (optimal solution) of the learning algorithm.

5.5 Comparison with Related Models

We highlight the difference between the proposed method and the closely related models, such as triangle divergence (Han et al. 2019) and cooperative network (Xie et al. 2018b). The proposed model optimizes

$$\min_{\theta,\alpha,\beta} \mathrm{KL}(\Pi||P) + \mathrm{KL}(P||Q)$$

or equivalently

$$\min_{\theta,\alpha,\beta} \mathrm{KL}(p_{\text{data}}||p_\theta) + \mathrm{KL}(p_\theta||q_\alpha) + \mathrm{KL}(\pi_\beta(z|x)||q_\alpha(z|x)),$$

which is different from the triangle divergence (Han et al. 2019) framework which also trains energy-based model, inference model and latent variable model together but optimizes the following different objective

$$\min_{\theta,\alpha,\beta} \mathrm{KL}(\Pi||Q) + \mathrm{KL}(Q||P) - \mathrm{KL}(\Pi||P).$$

The cooperative learning (Xie et al. 2018b) framework (CoopNets) jointly trains the energy-based model $p_\theta(x)$ and the latent variable model $q_\alpha(x)$ by

$$\min_{\theta,\alpha} \mathrm{KL}(p_{\text{data}}||p_\theta) + \mathrm{KL}(p_\theta||q_\alpha),$$

without leaning an approximate $\pi_\beta(z|x)$. Instead, Coop-Nets (Xie et al. 2018b) directly accesses the inference process $q_\alpha(z|x)$ by MCMC sampling.

6 Experiments

We present experiments to demonstrate the effectiveness of our strategy to train EBMs with (a) competitive synthesis for images, (b) high expressiveness of the learned latent variable model, and (c) strong performance in image completion. We use the PaddlePaddle [1] deep learning platform.

6.1 Image Generation

We show that our framework is effective to represent a probability density of images. We demonstrate the learned model can generate realistic image patterns. We learn our model from MNIST (LeCun et al. 1998), Fashion-MNIST (Xiao, Rasul, and Vollgraf 2017) and CIFAR-10 (Krizhevsky 2009) images without class labels. Figure 6 shows some examples generated by the *ancestral Langevin sampling*. We also quantitatively evaluate the qualities of the generated images via FID score (Heusel et al. 2017) and Inception score (Salimans et al. 2016) in Table 1 and Table 2. The experiments validate the effectiveness of our model. We design all networks in our model with simple convolution and ReLU layers, and only use 15 or 50 Langevin steps. The Langevin step size $\delta = 0.002$. The number of latent dimension $d = 200$.

Model	FID
GLO (Bojanowski et al. 2018)	49.60
CGlow (Liu et al. 2019)	29.64
CAGlow (Liu et al. 2019)	26.34
VAE (Kingma and Welling 2014)	21.85
DDGM (Kim and Bengio 2016)	30.87
BEGAN (Berthelot, Schumm, and Metz 2017)	13.54
EBGAN (Zhao, Mathieu, and LeCun 2017)	11.10
Triangle (Han et al. 2019)	6.77
CoopNets (Xie et al. 2018b)	9.70
Ours	8.95

Table 1: Comparison with baseline models on MNIST dataset with respect to FID score ($l = 50$).

We also check whether the latent variable model $q_\alpha(x|z)$ learns a meaningful latent space z in this learning scheme by demonstrating interpolation between generated examples in the latent space as shown in Figure 7(a). Each row of transition is a sequence of $g_\alpha(z_\eta)$ with interpolated $z_\eta = \eta z_l + \sqrt{1 - \eta^2} z_r$ where $\eta \in [0, 1]$, z_l and z_r are the latent variables of the examples at the left and right ends respectively. The transitions appear smooth, which means that the latent variable model learns a meaningful image embedding.

[1]https://www.paddlepaddle.org.cn

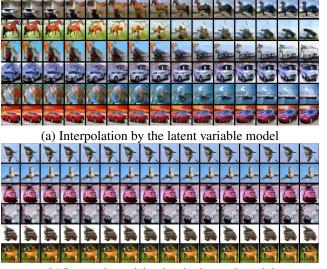Figure 6: Generated samples by the models learned on MNIST, Fashion-MNIST and CIFAR-10 datasets respectively.

Model	IS
PixelCNN (Van den Oord et al. 2016)	4.60
PixelIQN (Ostrovski, Dabney, and Munos 2018)	5.29
EBM (Du and Mordatch 2019)	6.02
DCGAN (Radford, Metz, and Chintala 2016)	6.40
WGAN+GP (Gulrajani et al. 2017)	6.50
CoopNets (Xie et al. 2018a)	6.55
Ours	6.65

Table 2: Inception scores on CIFAR-10 dataset ($l = 15$).

We also check the gap between p_θ and q_α once the model is leaned, by visualizing the Langevin dynamics initialized by a sample from the latent variable model in Figure 7(b). Each row shows one example, in which the leftmost image is generated by the latent variable model via ancestral sampling, and the rest image sequence shows the revised examples at different Langevin steps. The rightmost one is the final synthesized example after 15 steps of Langevin revision. We can find that even though the Langevin dynamics can still improve the initial example (we can carefully compare the leftmost and the rightmost images, the rightmost one is a little bit sharper than the leftmost one), but their difference is quite small, which is in fact a good phenomenon revealing that the latent variable model has caught up with the EBM, which runs toward the data distribution. That is, q_α becomes the stationary distribution of p_θ, or $\text{KL}(\mathcal{M}_{\hat\theta} q_{\hat\alpha}(x) || q_{\hat\alpha}(x)) \to 0$.

6.2 Image Completion

We apply our conditional model to image completion, where we learn a stochastic mapping from a centrally masked image to the original one. The centrally masked image is of the size 256×256 pixels, centrally overlaid with a mask of the size 128×128 pixels. The conditional energy function in $p_\theta(x|y)$ takes the concatenation of the masked image y and the original image x as input and consists of three convolutional layers and one fully-connected layer. For the conditional latent variable model $q_\alpha(x|y, z)$, we follow Isola et al. (2017) to use a U-Net (Ronneberger, Fischer, and Brox 2015), with the latent vector z concatenated

(a) Interpolation by the latent variable model

(b) Langevin revision by the learned model

Figure 7: Model analysis. (a) Interpolation between latent vectors of the images on the two ends. (b) Visualization of ancestral Langevin dynamics when the model converges. For each row, the leftmost image is the synthesized output by the ancestral sampling. The rest image sequence displays the synthesized images revised at different Langevin steps.

with its bottleneck. We set $d = 8$. The conditional encoder $\pi_\beta(z|y, x)$ has five residual blocks and MLP layers to get the variational encoding. We compare our method with baselines including pix2pix (Isola et al. 2017), cVAE-GAN (Zhu et al. 2017), cVAE-GAN++ (Zhu et al. 2017), Bicycle-GAN (Zhu et al. 2017), and cCoopNets (Xie et al. 2018a) on the Paris StreetView (Pathak et al. 2016) and the CMP Facade datasets (Tyleček and Šára 2013) in Table 3. The recovery performance is measured by the peak signal-to-noise ratio (PSNR) and Structural SIMilarity (SSIM) between the recovered image and the original image. Our method outperforms the baselines. Figure 8 shows some qualitative results.
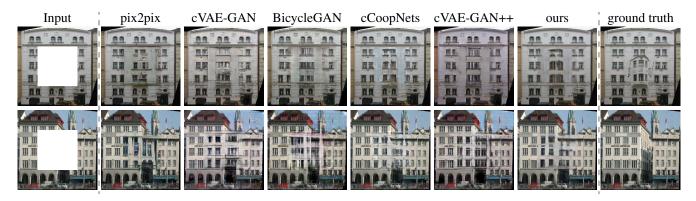
Figure 8: Example results of image completion on the Facade test dataset.

Method	Facade		StreetView	
	PSNR	SSIM	PSNR	SSIM
pix2pix	19.34	0.74	15.17	0.75
cVAE-GAN	19.43	0.68	16.12	0.72
cVAE-GAN++	19.14	0.64	16.03	0.69
BicycleGAN	19.07	0.64	16.00	0.68
cCoopNets	20.47	0.77	21.17	0.79
Ours	**21.62**	**0.78**	**22.61**	**0.79**

Table 3: Comparison with baselines for image completion.

6.3 Model Analysis

Our framework involves three different components, each of which has some key hyper-parameters that might affect the behavior of the whole training process. We investigate some factors that may potentially influence the performance of our framework on CIFAR-10. The results are reported after 1,000 epochs of training.

Number of Langevin steps and Lagevin step size We first study how the number of Langevin steps and their step size affect the synthesis performance. Table 4 shows the influence of varying number of Langevin step and Langevin step size, respectively. As the number of Langevin steps increases and the step size decreases, we observe improved quality of image synthesis in terms of inception score.

IS ↑	$l = 5$	$l = 8$	$l = 15$	$l = 30$	$l = 60$
$\delta = 0.001$	3.606	4.333	6.072	6.038	6.143
$\delta = 0.002$	3.847	5.568	6.075	5.989	5.882
$\delta = 0.004$	4.799	5.286	5.979	5.907	5.933
$\delta = 0.008$	5.146	5.164	5.835	4.574	3.482

Table 4: Influence of number of MCMC steps l and MCMC step size δ, with the number of latent dimension $d = 200$, and variational loss penalty $\gamma = 2$.

Number of dimensions of the latent space We also study how the number of dimensions of the latent space affect the *ancestral Langevin sampling* process in training the energy-based model. Table 5 displays the inception scores as a func-

tion of the number of latent dimensions of $q_\alpha(x)$. We set $l = 10$, $\delta = 0.002$, and $\gamma = 2$.

d	1200	600	200	100	50	10
IS ↑	6.017	6.213	6.159	6.085	6.027	5.973

Table 5: Influence of the number of latent dimension d

Variational loss penalty The penalty weight γ of the term of KL-divergence between the inference model and the posterior distribution in Eq. (9) plays an important role in adjusting the tradeoff between having low auto-encoding reconstruction loss and having good approximation of the posterior distribution. Table 6 displays the inception scores of varying γ, with $d = 200$, $l = 10$, and $\delta = 0.002$. The optimal choice of γ in our model is roughly 2.

γ	0.05	0.5	1	2	8	10
IS ↑	5.106	5.663	5.905	6.159	5.890	4.693

Table 6: Influence of the variational loss penalty γ

7 Conclusion

This paper proposes to learn an EBM with a VAE as an amortized sampler for probability density estimation. In particular, we propose the *variational MCMC teaching* algorithm to train the EBM and VAE together. In the proposed joint training framework, the latent variable model in the VAE and the Langevin dynamics derived from the EBM learn to collaborate to form an efficient sampler, which is essential to provide Monte Carlo samples to train both the EBM and the VAE. The proposed method naturally unifies the maximum likelihood estimation, variational learning, and MCMC teaching in a single computational framework, and can be interpreted as a dynamic alternating projection within the framework of information geometry. Our framework is appealing as it combines the representational flexibility and ability of the EBM and the computational tractability and efficiency of the VAE. Experiments show that the proposed framework can be effective in image generation, and its conditional generalization can be useful for computer vision applications, such as image completion.

References

Bakhtin, A.; Deng, Y.; Gross, S.; Ott, M.; Ranzato, M.; and Szlam, A. 2021. Residual energy-Based models for text. *Journal of Machine Learning Research (JMLR)* 22(40): 1–41.

Barbu, A.; and Zhu, S.-C. 2020. *Monte Carlo methods.* Springer.

Berthelot, D.; Schumm, T.; and Metz, L. 2017. BE-GAN: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717* .

Bojanowski, P.; Joulin, A.; Lopez-Pas, D.; and Szlam, A. 2018. Optimizing the latent space of generative networks. In *International Conference on Machine Learning (ICML)*, 600–609.

Cover, T. M. 1999. *Elements of Information Theory*. John Wiley & Sons.

Du, Y.; Meier, J.; Ma, J.; Fergus, R.; and Rives, A. 2019. Energy-based models for atomic-resolution protein conformations. In *International Conference on Learning Representations (ICLR)*.

Du, Y.; and Mordatch, I. 2019. Implicit generation and modeling with energy based models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 3608–3618.

Gao, R.; Lu, Y.; Zhou, J.; Zhu, S.-C.; and Nian Wu, Y. 2018. Learning generative convnets via multi-grid modeling and sampling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 9155–9164.

Geman, S.; and Geman, D. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 6(6): 721–741.

Grathwohl, W.; Wang, K.-C.; Jacobsen, J.-H.; Duvenaud, D.; Norouzi, M.; and Swersky, K. 2019. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations (ICLR)*.

Grenander, U.; and Miller, M. I. 2007. *Pattern Theory: From Representation to Inference*. Oxford University Press.

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. C. 2017. Improved training of wasserstein GANs. In *Advances in Neural Information Processing Systems (NIPS)*, 5767–5777.

Gutmann, M. U.; and Hyvärinen, A. 2012. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research (JMLR)* 13(2): 307–361.

Han, T.; Lu, Y.; Zhu, S.-C.; and Wu, Y. N. 2017. Alternating back-propagation for generator network. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI)*, 1976–1984.

Han, T.; Nijkamp, E.; Fang, X.; Hill, M.; Zhu, S.-C.; and Wu, Y. N. 2019. 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 (CVPR)*, 8670–8679.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems (NIPS)*, 6626–6637.

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

Hinton, G. E. 2012. A practical guide to training restricted Boltzmann machines. In *Neural Networks: Tricks of the Trade*, 599–619.

Ingraham, J.; Riesselman, A.; Sander, C.; and Marks, D. 2018. Learning protein structure with a differentiable simulator. In *International Conference on Learning Representations (ICLR)*.

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1125–1134.

Kim, T.; and Bengio, Y. 2016. Deep directed generative models with energy-based probability estimation. *arXiv preprint arXiv:1606.03439* .

Kingma, D. P.; and Ba, J. 2015. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)* .

Kingma, D. P.; and Welling, M. 2014. Auto-encoding variational Bayes. In *International Conference on Learning Representations (ICLR)*.

Krizhevsky, A. 2009. Learning multiple layers of features from tiny images. Technical report, University of Toronto.

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

LeCun, Y.; Chopra, S.; Hadsell, R.; Ranzato, M.; and Huang, F. J. 2006. A tutorial on energy-based learning. In *Predicting Structured Data*. MIT Press.

Liu, J. S. 2008. *Monte Carlo strategies in scientific computing*. Springer Science & Business Media.

Liu, R.; Liu, Y.; Gong, X.; Wang, X.; and Li, H. 2019. Conditional adversarial generative flow for controllable image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7992–8001.

Lu, Y.; Zhu, S.-C.; and Wu, Y. N. 2016. Learning FRAME models using CNN filters. In *Thirtieth AAAI Conference on Artificial Intelligence (AAAI)*, 1902–1910.

Neal, R. M. 2011. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo* 2.

Nijkamp, E.; Hill, M.; Zhu, S.-C.; and Wu, Y. N. 2019. Learning non-convergent non-persistent short-run MCMC toward energy-based model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 5233–5243.

Ostrovski, G.; Dabney, W.; and Munos, R. 2018. Autoregressive quantile networks for generative modeling. In *International Conference on Machine Learning (ICML)*, 3936–3945.

Pathak, D.; Krahenbuhl, P.; Donahue, J.; Darrell, T.; and Efros, A. A. 2016. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2536–2544.

Radford, A.; Metz, L.; and Chintala, S. 2016. Unsupervised representation learning with deep convolutional generative adversarial networks. *International Conference on Learning Representations (ICLR)* .

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, 234–241.

Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; and Chen, X. 2016. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems (NIPS)*, 2226–2234.

Sohn, K.; Lee, H.; and Yan, X. 2015. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems (NIPS)*, 3483–3491.

Song, Y.; and Ou, Z. 2018. Learning neural random fields with inclusive auxiliary generators. *arXiv preprint arXiv:1806.00271* .

Tyleček, R.; and Šára, R. 2013. Spatial pattern templates for recognition of objects with regular structure. In *German Conference on Pattern Recognition (GCPR)*, 364–374.

Van den Oord, A.; Kalchbrenner, N.; Espeholt, L.; Vinyals, O.; and Graves, A. 2016. Conditional image generation with pixelCNN decoders. In *Advances in Neural Information Processing Systems (NIPS)*, 4790–4798.

Wu, Y. N.; Zhu, S.-C.; and Liu, X. 2000. Equivalence of Julesz ensembles and FRAME models. *International Journal of Computer Vision (IJCV)* 38: 247–265.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747* .

Xie, J.; Hu, W.; Zhu, S.-C.; and Wu, Y. N. 2014. Learning sparse FRAME models for natural image patterns. *International Journal of Computer Vision (IJCV)* 1–22.

Xie, J.; Lu, Y.; Gao, R.; and Wu, Y. N. 2018a. Cooperative learning of energy-based model and latent variable model via MCMC teaching. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 4292–4301.

Xie, J.; Lu, Y.; Gao, R.; Zhu, S.-C.; and Wu, Y. N. 2018b. Cooperative training of descriptor and generator networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 42(1): 27–45.

Xie, J.; Lu, Y.; Zhu, S.-C.; and Wu, Y. 2016. A theory of generative ConvNet. In *International Conference on Machine Learning (ICML)*, 2635–2644.

Xie, J.; Xu, Y.; Zheng, Z.; Zhu, S.; and Wu, Y. N. 2021a. Generative PointNet: energy-based learning on unordered point sets for 3D generation, reconstruction and classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xie, J.; Zheng, Z.; Fang, X.; Zhu, S.-C.; and Wu, Y. N. 2019. Cooperative training of fast thinking initializer and slow thinking solver for multi-modal conditional learning. *arXiv preprint arXiv:1902.02812* .

Xie, J.; Zheng, Z.; Fang, X.; Zhu, S.-C.; and Wu, Y. N. 2021b. Learning cycle-consistent cooperative networks via alternating MCMC teaching for unsupervised cross-domain translation. In *Proceedings of The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI)*.

Xie, J.; Zheng, Z.; Gao, R.; Wang, W.; Zhu, S.-C.; and Wu, Y. N. 2018c. Learning descriptor networks for 3D shape synthesis and analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 8629–8638.

Xie, J.; Zheng, Z.; Gao, R.; Wang, W.; Zhu, S.-C.; and Wu, Y. N. 2020. Generative VoxelNet:learning energy-based models for 3D shape synthesis and analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* .

Xie, J.; Zhu, S.-C.; and Wu, Y. N. 2017. Synthesizing dynamic patterns by spatial-temporal generative ConvNet. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7093–7101.

Xie, J.; Zhu, S.-C.; and Wu, Y. N. 2019. Learning energy-based spatial-temporal generative convnets for dynamic patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* .

Xu, Y.; Xie, J.; Zhao, T.; Baker, C.; Zhao, Y.; and Wu, Y. N. 2019. Energy-based continuous inverse optimal control. *arXiv preprint arXiv:1904.05453* .

Zhao, J.; Mathieu, M.; and LeCun, Y. 2017. Energy-based generative adversarial networks. In *International Conference on Learning Representations (ICLR)*.

Zhu, J.-Y.; Zhang, R.; Pathak, D.; Darrell, T.; Efros, A. A.; Wang, O.; and Shechtman, E. 2017. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems (NIPS)*, 465–476.

Zhu, S. C.; Wu, Y.; and Mumford, D. 1998. Filters, random fields and maximum entropy (FRAME): Towards a unified theory for texture modeling. *International Journal of Computer Vision (IJCV)* 27(2): 107–126.