Learning Latent Space Energy-Based Prior Model

Bo Pang[*]
Department of Statistics
UCLA
bopang@ucla.edu

Tian Han[*]
Department of Computer Science
Stevens Institute of Technology
than6@stevens.edu

Erik Nijkamp[*]
Department of Statistics
UCLA
enijkamp@ucla.edu

Song-Chun Zhu
Department of Statistics
UCLA
sczhu@stat.ucla.edu

Ying Nian Wu
Department of Statistics
UCLA
ywu@stat.ucla.edu

Abstract

The generator model assumes that the observed example is generated by a low-dimensional latent vector via a top-down network, and the latent vector follows a simple and known prior distribution, such as uniform or Gaussian white noise distribution. While we can learn an expressive top-down network to map the prior distribution to the data distribution, we can also learn an expressive prior model instead of assuming a given prior distribution. This follows the philosophy of empirical Bayes where the prior model is learned from the observed data. We propose to learn an energy-based prior model for the latent vector, where the energy function is parametrized by a very simple multi-layer perceptron. Due to the low-dimensionality of the latent space, learning a latent space energy-based prior model proves to be both feasible and desirable. In this paper, we develop the maximum likelihood learning algorithm and its variation based on short-run Markov chain Monte Carlo sampling from the prior and the posterior distributions of the latent vector, and we show that the learned model exhibits strong performance in terms of image and text generation and anomaly detection.

1 Introduction

In recent years, deep generative models have achieved impressive successes in image and text generation. A particularly simple and powerful model is the generator model [29, 18], which maps a low-dimensional latent vector to image or text via a top-down network. The generator model was proposed in the contexts of variational auto-encoder (VAE) [29, 48] and generative adversarial networks (GAN) [18, 47]. In both frameworks, the generator model is jointly learned with a complementary model, such as the inference model in VAE and the discriminator model in GAN. More recently in [20, 44], the generator model has also been learned by maximum likelihood without resorting to a complementary model, where the inference is carried out by Markov chain Monte Carlo (MCMC) such as the Langevin dynamics. In this paper, we shall adopt the the framework of maximum likelihood estimate (MLE), instead of GAN or VAE, so that the learning is simpler in the sense that we do not need to train a complementary network.

The expressive power of the generator network for image and text generation comes from the top-down network that maps a simple prior distribution to be close to the data distribution. Most of the existing papers [39, 54, 2, 13, 57, 31] assume that the latent vector follows a given simple prior distribution, such as isotropic Gaussian white noise distribution or uniform distribution. However,

[*]Equal contributions.

Preprint. Under review.

such assumption may cause ineffective generator learning as observed in [11, 56]. While we can increase the complexity of the top-down network to enhance the expressive power of the model, in this paper, we shall pursue a different strategy by following the philosophy of empirical Bayes, that is, instead of assuming a given prior distribution for the latent vector, we learn a prior model from empirical observations.

Figure 1: Generated images for CelebA $(128 \times 128 \times 3)$.

Specifically, we assume the latent vector follows an energy-based model (EBM), or more specifically, an energy-based correction of the isotropic Gaussian white noise prior distribution. We call this model the latent space energy-based prior model. Such a prior model adds more expressive power to the generator model.

The MLE learning of the generator model with a latent space EBM prior involves MCMC sampling of latent vector from both the prior and posterior distributions. Parameters of the prior model can then be updated based on the statistical difference between samples from the two distributions. Parameters of the top-down network can be updated based on the samples from the posterior distribution as well as the observed data.

Compared to GAN that involves delicate dueling between two networks, MLE learning is simpler, and does not suffer from issues such as instability or mode collapsing. As to VAE, for generator model with a latent space EBM prior, VAE is not easily applicable because of the intractability of the normalizing constant of the latent space EBM.

Although MLE learning does not require training a complementary model, it requires MCMC sampling from prior and posterior distributions of the learned model. However, because MCMC sampling is carried out in the low-dimensional latent space instead of the high-dimensional data space, it is easily affordable on modern computing platforms. Compared to EBM built directly on image or text, the latent space EBM prior can be much less multi-modal, because it can rely on the top-down network to map the prior distribution to the highly multi-modal data distribution. A less multi-modal EBM is more amendable to MCMC sampling.

Furthermore, in this paper, we propose to use short-run MCMC sampling [43, 42, 45], i.e., we always initialize MCMC from the fixed Gaussian white noise distribution, and we always run a fixed and small number of steps, in both training and testing stages. Such a learning algorithm is simple and efficient. We formulate this learning algorithm as a perturbation of MLE learning in terms of both objective function and estimating equation, so that the learning algorithm has a soild theoretical foundation.

We test the proposed modeling, learning and computing method on tasks such as image synthesis, text generation, as well as anomaly detection. We show that our method is competitive with prior art.

Contributions. (1) We propose a generator model with a latent space energy-based prior model by following the empirical Bayes philosophy. (2) We develop the maximum likelihood learning algorithm based on MCMC sampling of the latent vector from the prior and posterior distributions. (3) We further develop an efficient modification of MLE learning based on short-run MCMC sampling [43, 42, 45].

(4) We provide theoretical foundation for learning driven by short-run MCMC. (5) We provide strong empirical results to corroborate the proposed method.

2 Modeling strategies and related work

We now put our work within the bigger picture of modeling and learning, and discuss related work.

Energy-based model and top-down generation model. A top-down model or a directed acyclic graphical (DAG) model is of a simple factorized form that is capable of ancestral sampling. The prototype of such a model is factor analysis [50], which has been generalized to independent component analysis [26], sparse coding [46], non-negative matrix factorization [34], etc. An early example of a multi-layer top-down model is the generation model of Helmholtz machine [24]. An energy-based model defines an unnormalized density or a Gibbs distribution. The prototype of such a model is exponential family distribution, the Boltzmann machine [1], and the FRAME (Filters, Random field, And Maximum Entropy) model [66]. [64] contrasted these two classes of models, calling the top-down latent variable model the generative model, and the energy-based model the descriptive model. [19] proposed to integrate the two models, where the top-down generation model generates textons, while the EBM prior accounts for the spatial placement and arrangement of textons. Our model follows such a scheme.

The energy function in the EBM can be viewed as the objective function, the cost function, the constraints, or a critic [51]. It is easy to specify, although optimizing or sampling the energy function can be hard, and may require iterative algorithm such as MCMC. The maximum likelihood learning of EBM can be interpreted as an adversarial scheme [58, 22, 33, 60], where the MCMC serves as the generator and the energy function serves as an evaluator. However, unlike GAN, the maximum likelihood learning of EBM does not suffer from issues such as mode collapsing.

The top-down generation model can be viewed as an actor [51] that directly generates the samples. It is easy to sample from, although one may need a complex top-down model to generate high quality samples. Comparing the two models, the EBM can be more expressive than a top-down model of the same complexity, while a top-down model is much easier to sample from. Therefore, it is desirable to let EBM take over the top layers of the top-down model to make the model more expressive, while EBM learning is still feasible.

Energy-based correction of top-down model. The top-down model usually assumes independent nodes at the top layer and conditional independent nodes at subsequent layers. We can introduce energy terms at multiple layers to correct for the independence or conditional independence assumptions. This leads to a latent energy-based model. However, unlike undirected latent EBM, the energy-based correction is learned on top of a directed top-down model, and this can be easier than learning an undirected latent EBM from scratch. Our work is a simple example of this scheme where we correct the prior distribution. We can also correct the generation model.

From data space EBM to latent space EBM. EBM learned in data space such as image space [59, 37, 21, 43, 15] can be highly multi-modal, and MCMC sampling can be difficult. In that case, we can introduce latent variables and learn an EBM in latent space, while also learning a mapping from the latent space to the data space. Our work follows such a strategy. Earlier papers on this strategy are [64, 19, 4, 8, 31]. Learning EBM in latent space can be much feasible than learning EBM in data space in terms of MCMC sampling, and much of past work on EBM can be re-casted in the latent space.

Short-run MCMC. Recently, [43] proposed to use short-run MCMC to sample from the EBM in data space. [45] proposed to use short-run MCMC to sample the latent variables of a top-down generation model from their posterior distribution. Our work adopts short-run MCMC to sample from both the prior and the posterior of the latent variables. We also provide theoretical foundation for the learning algorithm with short-run MCMC sampling.

Generator model with flexible prior. A few variants of VAE attempt to address the mismatch between the prior and the aggregate posterior. VampPrior [55] parameterizes the prior based on the posterior inference model, while [3] proposes to construct rich priors using rejection sampling with a learned acceptance function, both yielding improved performance on grey scale images. ARAE [63] learns an implicit prior distribution in the latent space with adversarial training and demonstrates superior performance on text generation. Recently, some papers resort to a two-stage approach

[12, 16]. They first train a VAE or deterministic autoencoder (with some form of regularization) in the data space. To enable generation from the model, they then fit a VAE or Gaussian mixture to the posterior samples inferred by the first stage model. An earlier model related to the two-stage approach is GLO [5] where a generator is trained paired with inference conducted by gradient descent on the latent variables instead of a separate inference network. All of these prior models by and large follow the empirical Bayes philosophy, which is also one motivation of our work.

3 Model and algorithm

3.1 Model

Let $x \in \mathbb{R}^D$ be an observed example such as an image or a piece of text, and let $z \in \mathbb{R}^d$ be the latent variables, where $D \gg d$. The joint distribution of (x, z) is

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

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

The prior model $p_\alpha(z)$ is formulated as an energy-based model,

$$p_\alpha(z) = \frac{1}{Z(\alpha)} \exp(f_\alpha(z)) p_0(z). \tag{2}$$

where $p_0(z)$ is a reference distribution, assumed to be isotropic Gaussian in this paper. $f_\alpha(z)$ is the negative energy and is parameterized by a small multi-layer perceptron with parameters α. $Z(\alpha) = \int \exp(f_\alpha(z)) p_0(z) dz = \mathrm{E}_{p_0}[\exp(f_\alpha(z))]$ is the normalizing constant or partition function.

The prior model (2) can be interpreted as an energy-based correction or exponential tilting of the original prior distribution p_0, which is the prior distribution in the generator model in VAE.

The generation model is the same as the top-down network in VAE. For image modeling,

$$x = g_\beta(z) + \epsilon, \tag{3}$$

where $\epsilon \sim \mathrm{N}(0, \sigma^2 I_D)$, so that $p_\beta(x|z) \sim \mathrm{N}(g_\beta(z), \sigma^2 I_D)$. As in VAE, σ^2 takes an assumed value. For text modeling, let $x = (x^{(t)}, t = 1, ..., T)$ where each $x^{(t)}$ is a token. Following previous text VAE model [7], we define $p_\beta(x|z)$ as a conditional autoregressive model,

$$p_\beta(x|z) = \prod_{t=1}^{T} p_\beta(x^{(t)}|x^{(1)}, ..., x^{(t-1)}, z) \tag{4}$$

which is often parameterized by a recurrent network with parameters β.

In the original generator model, the top-down network g_β maps the unimodal prior distribution p_0 to be close to the usually highly multi-modal data distribution. The prior model in (2) refines p_0 so that g_β maps the prior model p_α to be closer to the data distribution. The prior model p_α does not need to be highly multi-modal because of the expressiveness of g_β.

The marginal distribution is $p_\theta(x) = \int p_\theta(x, z) dz = \int p_\alpha(z) p_\beta(x|z) dz$. The posterior distribution is $p_\theta(z|x) = p_\theta(x, z)/p_\theta(x) = p_\alpha(z) p_\beta(x|z)/p_\theta(x)$.

In the above model, we exponentially tilt $p_0(z)$. We can also exponentially tilt $p_0(x, z) = p_0(z) p_\beta(x|z)$ to $p_\theta(x, z) = \frac{1}{Z(\theta)} \exp(f_\alpha(x, z)) p_0(x, z)$. Equivalently, we may also exponentially tilt $p_0(z, \epsilon) = p_0(z) p(\epsilon)$, as the mapping from (z, ϵ) to (z, x) is a change of variable. This leads to an EBM in both the latent space and data space, which makes learning and sampling more complex. Therefore, we choose to only tilt $p_0(z)$ and leave $p_\beta(x|z)$ as a directed top-down generation model.

3.2 Maximum likelihood

Suppose we observe training examples $(x_i, i = 1, ..., n)$. The log-likelihood function is

$$L(\theta) = \sum_{i=1}^{n} \log p_\theta(x_i). \tag{5}$$

The learning gradient can be calculated according to

$$\nabla_\theta \log p_\theta(x) = \mathrm{E}_{p_\theta(z|x)}\left[\nabla_\theta \log p_\theta(x,z)\right] = \mathrm{E}_{p_\theta(z|x)}\left[\nabla_\theta(\log p_\alpha(z) + \log p_\beta(x|z))\right]. \quad (6)$$

See Appendix 6.3 and 6.4 for a detailed derivation.

For the prior model, $\nabla_\alpha \log p_\alpha(z) = \nabla_\alpha f_\alpha(z) - \mathrm{E}_{p_\alpha(z)}[\nabla_\alpha f_\alpha(z)]$. Thus the learning gradient for an example x is

$$\delta_\alpha(x) = \nabla_\alpha \log p_\theta(x) = \mathrm{E}_{p_\theta(z|x)}[\nabla_\alpha f_\alpha(z)] - \mathrm{E}_{p_\alpha(z)}[\nabla_\alpha f_\alpha(z)]. \quad (7)$$

The above equation has an empirical Bayes nature. $p_\theta(z|x)$ is based on the empirical observation x, while p_α is the prior model. α is updated based on the difference between z inferred from empirical observation x, and z sampled from the current prior.

For the generation model,

$$\delta_\beta(x) = \nabla_\beta \log p_\theta(x) = \mathrm{E}_{p_\theta(z|x)}[\nabla_\beta \log p_\beta(x|z)], \quad (8)$$

where $\log p_\beta(x|z) = -\|x - g_\beta(z)\|^2/(2\sigma^2) + \text{constant}$ or $\sum_{t=1}^{T} \log p_\beta(x^{(t)}|x^{(1)}, ..., x^{(t-1)}, z)$ for image and text modeling respectively, which is about the reconstruction error.

Expectations in (7) and (8) require MCMC sampling of the prior model $p_\alpha(z)$ and the posterior distribution $p_\theta(z|x)$. We can use Langevin dynamics [32, 65]. For a target distribution $\pi(z)$, the dynamics iterates

$$z_{k+1} = z_k + s\nabla_z \log \pi(z_k) + \sqrt{2s}\epsilon_k, \quad (9)$$

where k indexes the time step of the Langevin dynamics, s is a small step size, and $\epsilon_k \sim \mathrm{N}(0, I_d)$ is the Gaussian white noise. $\pi(z)$ can be either $p_\alpha(z)$ or $p_\theta(z|x)$. In either case, $\nabla_z \log \pi(z)$ can be efficiently computed by back-propagation.

It is worth noting that VAE is not conveniently applicable here. Even if we have a tractable approximation to $p_\theta(z|x)$ in the form of an inference network, we still need to compute $\mathrm{E}_{p_\alpha(z)}[\nabla_\alpha f_\alpha(z)]$, which requires MCMC.

3.3 Short-run MCMC

Convergence of Langevin dynamics to the target distribution requires infinite steps with infinitesimal step size, which is impractical. We thus propose to use short-run MCMC [43, 42, 45] for approximate sampling. This is in agreement with the philosophy of variational inference, which accepts the intractability of the target distribution and seeks to approximate it by a simpler distribution. The difference is that we adopt short-run Langevin dynamics instead of learning a separate network for approximation.

The short-run Langevin dynamics is always initialized from the fixed initial distribution p_0, and only runs a fixed number of K steps, e.g., $K = 20$,

$$z_0 \sim p_0(z),\ z_{k+1} = z_k + s\nabla_z \log \pi(z_k) + \sqrt{2s}\epsilon_k,\ k = 1, ..., K. \quad (10)$$

Denote the distribution of z_K to be $\tilde{\pi}(z)$. Because of fixed $p_0(z)$ and fixed K and s, the distribution $\tilde{\pi}$ is well defined. In this paper, we put ~ sign on top of the symbols to denote distributions or quantities produced by short-run MCMC, and for simplicity, we omit the dependence on K and s in notation. As shown in [10], the Kullback-Leibler divergence $D_{KL}(\tilde{\pi}\|\pi)$ decreases to zero monotonically as $K \to \infty$.

Specifically, denote the distribution of z_K to be $\tilde{p}_\alpha(z)$ if the target $\pi(z) = p_\alpha(z)$, and denote the distribution of z_K to be $\tilde{p}_\theta(z|x)$ if $\pi(z) = p_\theta(z|x)$. We can then replace $p_\alpha(z)$ by $\tilde{p}_\alpha(z)$ and replace $p_\theta(z|x)$ by $\tilde{p}_\theta(z|x)$ in equations (7) and (8), so that the learning gradients in equations (7) and (8) are modified to

$$\tilde{\delta}_\alpha(x) = \mathrm{E}_{\tilde{p}_\theta(z|x)}[\nabla_\alpha f_\alpha(z)] - \mathrm{E}_{\tilde{p}_\alpha(z)}[\nabla_\alpha f_\alpha(z)], \quad (11)$$

$$\tilde{\delta}_\beta(x) = \mathrm{E}_{\tilde{p}_\theta(z|x)}[\nabla_\beta \log p_\beta(x|z)]. \quad (12)$$

We then update α and β based on (62) and (63), where the expectations can be approximated by Monte Carlo samples.

The short-run MCMC sampling is always initialized from the same initial distribution $p_0(z)$, and always runs a fixed number of K steps. This is the case for both training and testing stages, which share the same short-run MCMC sampling.

3.4 Algorithm

The learning and sampling algorithm is described in Algorithm 1.

Algorithm 1: Learning latent space EBM prior via short-run MCMC.

input : Learning iterations T, learning rate for prior model η_0, learning rate for generation model η_1, initial parameters $\theta_0 = (\alpha_0, \beta_0)$, observed examples $\{x_i\}_{i=1}^n$, batch size m, number of prior and posterior sampling steps $\{K_0, K_1\}$, and prior and posterior sampling step sizes $\{s_0, s_1\}$.

output : $\theta_T = (\alpha_T, \beta_T)$.

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

1. **Mini-batch**: Sample observed examples $\{x_i\}_{i=1}^m$.
2. **Prior sampling**: For each x_i, sample $z_i^- \sim \tilde{p}_{\alpha_t}(z)$ using equation (10), where the target distribution $\pi(z) = p_{\alpha_t}(z)$, and $s = s_0$, $K = K_0$.
3. **Posterior sampling**: For each x_i, sample $z_i^+ \sim \tilde{p}_{\theta_t}(z|x_i)$ using equation (10), where the target distribution $\pi(z) = p_{\theta_t}(z|x_i)$, and $s = s_1$, $K = K_1$.
4. **Learning prior model**: $\alpha_{t+1} = \alpha_t + \eta_0 \frac{1}{m} \sum_{i=1}^m [\nabla_\alpha f_{\alpha_t}(z_i^+) - \nabla_\alpha f_{\alpha_t}(z_i^-)]$.
5. **Learning generation model**: $\beta_{t+1} = \beta_t + \eta_1 \frac{1}{m} \sum_{i=1}^m \nabla_\beta \log p_{\beta_t}(x_i|z_i^+)$.

The prior sampling and posterior sampling correspond to the positive phase and negative phase of latent EBM [1]. Learning prior model and learning generation model are based on mini-batch Monte Carlo approximations to (62) and (63) respectively.

3.5 Theoretical understanding

The learning algorithm based on short-run MCMC sampling in Algorithm 1 is a modification or perturbation of maximum likelihood learning, where we replace $p_\alpha(z)$ and $p_\theta(z|x)$ by $\tilde{p}_\alpha(z)$ and $\tilde{p}_\theta(z|x)$ respectively. For theoretical underpinning, we should also understand this perturbation in terms of objective function and estimating equation.

In terms of objective function, define the Kullback-Leibler divergence $D_{KL}(p(x)\|q(x)) = \mathrm{E}_p[\log(p(x)/q(x)]$. At iteration t, with fixed $\theta_t = (\alpha_t, \beta_t)$, consider the following perturbation of the log-likelihood function of θ for an observation x,

$$\log \tilde{p}_\theta(x) = \log p_\theta(x) - D_{KL}(\tilde{p}_{\theta_t}(z|x)\|p_\theta(z|x)) + D_{KL}(\tilde{p}_{\alpha_t}(z)\|p_\alpha(z)). \tag{13}$$

The above is a function of θ, while θ_t is fixed. Then

$$\tilde{\delta}_\alpha(x) = \nabla_\alpha \log \tilde{p}_\theta(x), \ \tilde{\delta}_\beta(x) = \nabla_\beta \log \tilde{p}_\theta(x), \tag{14}$$

where the derivative is taken at θ_t. See Appendix 6.8 for details. Thus the updating rule of Algorithm 1 follows the stochastic gradient (i.e., Monte Carlo approximation of the gradient) of a perturbation of the log-likelihood ($\tilde{p}_\theta(x)$ above is not necessarily a normalized density function any more). Equivalently, because θ_t is fixed, we can drop the entropies of $\tilde{p}_{\theta_t}(z|x)$ and $\tilde{p}_{\alpha_t}(z)$ in the above Kullback-Leibler divergences, hence the updating rule follows the stochastic gradient of

$$Q(\theta) = L(\theta) + \sum_{i=1}^n \left[\mathrm{E}_{\tilde{p}_{\theta_t}(z_i|x_i)}[\log p_\theta(z_i|x_i)] - \mathrm{E}_{\tilde{p}_{\alpha_t}(z)}[\log p_\alpha(z)]\right], \tag{15}$$

where $L(\theta)$ is the total log-likelihood defined in equation (5), and the gradient is taken at θ_t.

In equation (13), the first D_{KL} term is related to variational inference, although we do not learn a separate inference model. The second D_{KL} term is related to contrastive divergence [53], except that $\tilde{p}_{\alpha_t}(z)$ is initialized from $p_0(z)$. It serves to cancel the intractable $\log Z(\alpha)$ term.

In terms of estimating equation, the stochastic gradient descent in Algorithm 1 is a Robbins-Monro stochastic approximation algorithm [49] that solves the following estimating equation:

$$\frac{1}{n}\sum_{i=1}^n \tilde{\delta}_\alpha(x_i) = \frac{1}{n}\sum_{i=1}^n \mathrm{E}_{\tilde{p}_\theta(z_i|x_i)}[\nabla_\alpha f_\alpha(z_i)] - \mathrm{E}_{\tilde{p}_\alpha(z)}[\nabla_\alpha f_\alpha(z)] = 0, \tag{16}$$

$$\frac{1}{n}\sum_{i=1}^n \tilde{\delta}_\beta(x_i) = \frac{1}{n}\sum_{i=1}^n \mathrm{E}_{\tilde{p}_\theta(z_i|x_i)}[\nabla_\beta \log p_\beta(x_i|z_i)] = 0. \tag{17}$$

The solution to the above estimating equation defines an estimator of the parameters. Algorithm 1 converges to this estimator under the usual regularity conditions of Robbins-Monro [49]. If we replace $\tilde{p}_\alpha(z)$ by $p_\alpha(z)$, and $\tilde{p}_\theta(z|x)$ by $p_\theta(z|x)$, then the above estimating equation is the maximum likelihood estimating equation.

The above theoretical understanding in terms of objective function and estimating equation is more general than that of maximum likelihood, which is a special case where the number of steps $K \to \infty$ and the step size $s \to 0$ in the Langevin dynamics in equation (10). Our theoretical understanding is clearly more relevant in practice where we can only afford finite K with non-zero s.

As to the step size s, we currently treat it as a tuning parameter. s can be more formally optimized by maximizing Q in equation (15) or maximizing the average of $\log \tilde{p}_\theta(x_i)$ defined by equation (13). We may also allow different step sizes for different steps k in the short-run Langevin dynamics. We leave this issue to future investigations.

4 Experiments

We present a set of experiments which highlight the effectiveness of our proposed model with (1) excellent synthesis for both visual and textual data outperforming state-of-the-art baselines, (2) high expressiveness of the learned prior model for both data modalities, and (3) strong performance in anomaly detection.

For image data, we include SVHN [41], CelebA [36], and CIFAR-10 [30]. For text data, we include PTB [40], Yahoo [61], and SNLI [6]. We refer to Appendix 7.1 for details.

4.1 Image modeling

We evaluate the quality of the generated and reconstructed images. If the model is well-learned, the latent EBM $\pi_\alpha(z)$ will fit the generator posterior $p_\theta(z|x)$ which in turn renders realistic generated samples as well as faithful reconstructions. We compare our model with VAE [29] and SRI [44] which assume a fixed Gaussian prior distribution for the latent vector and two recent strong VAE variants, 2sVAE [12] and RAE [16], whose prior distributions are learned with posterior samples in a second stage. We also compare with multi-layer generator (i.e., 5 layers of latent vectors) model [44] which admits a powerful learned prior on the bottom layer of latent vector. We follow the protocol as in [44].

Generation. The generator network p_θ in our framework is well-learned to generate samples that are realistic and share visual similarities as the training data. The qualitative results are shown in Figure 2. We further evaluate our model quantitatively by using Fréchet Inception Distance (FID) [38] in Table 1. It can be seen that our model achieves superior generation performance compared to listed baseline models.

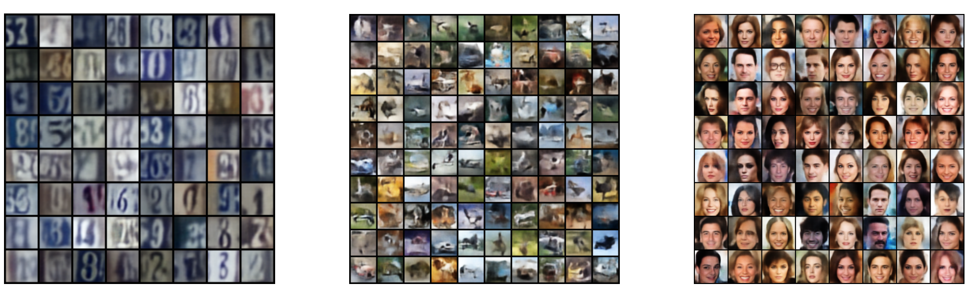

Figure 2: Generated samples for SVHN ($32 \times 32 \times 3$), CIFAR-10 ($32 \times 32 \times 3$), and CelebA ($64 \times 64 \times 3$).

Reconstruction. We then evaluate the accuracy of the posterior Langevin process by testing image reconstruction. The well-formed posterior Langevin should not only help to learn the latent EBM model but also learn to match the true posterior $p_\theta(z|x)$ of the generator model. We quantitatively compare reconstructions of test images with the above baseline models Mean Square Error (MSE). From Table 1, our proposed model could achieve not only high generation quality but also accurate reconstructions.

Models		VAE	2sVAE	RAE	SRI	SRI (L=5)	Ours
SVHN	MSE	0.019	0.019	0.014	0.018	0.011	**0.008**
	FID	46.78	42.81	40.02	44.86	35.23	**29.44**
CIFAR-10	MSE	0.057	0.056	0.027	-	-	**0.020**
	FID	106.37	109.77	74.16	-	-	**70.15**
CelebA	MSE	0.021	0.021	0.018	0.020	0.015	**0.013**
	FID	65.75	49.70	40.95	61.03	47.95	**37.87**

Table 1: MSE of testing reconstructions and FID of generated samples for SVHN $(32 \times 32 \times 3)$, CIFAR-10 $(32 \times 32 \times 3)$, and CelebA $(64 \times 64 \times 3)$ datasets.

4.2 Text modeling

We compare our model to related baselines, SA-VAE [27], FB-VAE [35], and ARAE [63]. SA-VAE involves optimizing posterior samples with gradient descent guided by EBLO, resembling the short run dynamics in our model. FB-VAE is the SOTA VAE for text modeling. While SA-VAE and FB-VAE assume a fixed Gaussian prior, ARAE learns a latent sample generator as an implicit prior distribution, which paired with a discriminator is adversarially trained. To evaluate the quality of the generated samples, we follow [63, 9] and recruit Forward Perplexity (FPPL) and Reverse Perplexity (RPPL). FPPL is the perplexity of the generated samples evaluated under a language model trained with real data and measures the fluency of the synthesized sentences. RPPL is the perplexity of real data (the test data partition) computed under a language model trained with the model-generated samples. Prior work employs it to measure the distributional coverage of a learned model, $p_\theta(x)$ in our case, since a model with a mode-collapsing issue results in a high RPPL. FPPL and RPPL are displayed in Table 2. Our model outperforms all the baselines on the two metrics, demonstrating the high fluency and diversity of the samples from our model. We also evaluate the reconstruction of our model against the baselines using negative log-likelihood (NLL). Our model has a similar performance as that of FB-VAE and ARAE, while they all outperform SA-VAE.

	SNLI			PTB			Yahoo		
Models	FPPL	RPPL	NLL	FPPL	RPPL	NLL	FPPL	RPPL	NLL
Real Data	23.53	-	-	100.36	-	-	60.04	-	-
SA-VAE	39.03	46.43	33.56	147.92	210.02	101.28	128.19	148.57	326.70
FB-VAE	39.19	43.47	28.82	145.32	204.11	92.89	123.22	141.14	319.96
ARAE	44.30	82.20	28.14	165.23	232.93	91.31	158.37	216.77	320.09
Ours	**27.81**	**31.96**	28.90	**107.45**	**181.54**	91.35	**80.91**	**118.08**	321.18

Table 2: Forward Perplexity (FPPL), Reverse Perplexity (RPPL), and Negative Log-Likelihood (NLL) for our model and baselines on SNLI, PTB, and Yahoo datasets.

4.3 Analysis of latent space

Short-run chains. We examine the exponential tilting of the reference prior $p_0(z)$ through Langevin samples initialized from $p_0(z)$ with target distribution $p_\alpha(z)$. As the reference distribution $p_0(z)$ is in the form of an isotropic Gaussian, we expect the energy-based correction f_α to tilt p_0 into an irregular shape. In particular, learning equation 62 may form shallow local modes for $p_\alpha(z)$. Therefore, the trajectory of a Markov chain initialized from the reference distribution $p_0(z)$ with well-learned target $p_\alpha(z)$ should depict the transition towards synthesized examples of high quality while the energy fluctuates around some constant. Figure 3 and Table 3 depict such transitions for image and textual data, respectively, which are both based on models trained with $K_0 = 40$ steps. For image data the quality of synthesis improve significantly with increasing number of steps. For textual data, there is an enhancement in semantics and syntax along the chain, which is especially clear from step 0 to 40 (see Table 3).

Long-run chains. While the learning algorithm 1 recruits short-run MCMC with K_0 steps to sample from target distribution $p_\alpha(z)$, a well-learned $p_\alpha(z)$ should allows for Markov chains with realistic synthesis for $K_0' \gg K_0$ steps. We demonstrate such long-run Markov chain with $K_0 = 40$ and $K_0' = 2500$ in Figure 4. The long-run chain samples in the data space are reasonable and do not exhibit the oversaturating issue of the long-run chain samples of recent EBM in the data space (see oversaturing examples in Figure 3 in [42]).

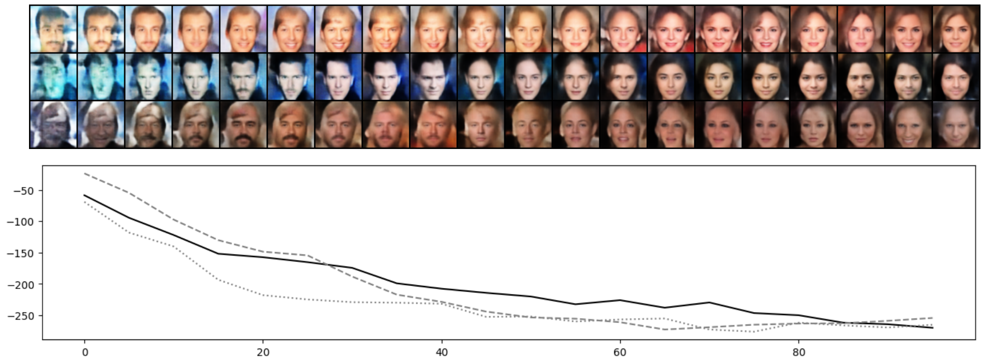

Figure 3: Transition of Markov chains initialized from $p_0(z)$ towards $\tilde{p}_\alpha(z)$ for $K_0' = 100$ steps. *Top:* Trajectory in the CelebA data-space. *Bottom:* Energy profile over time.

judge in <unk> was not
west virginia bank <unk> which has been under N law took effect of october N
mr. peterson N years old could return to work with his clients to pay
iras must be
anticipating bonds tied to the imperial company 's revenue of $ N million today
many of these N funds in the industrial average rose to N N from N N N
fund obtaining the the
ford 's latest move is expected to reach an agreement in principle for the sale of its loan operations
wall street has been shocked over by the merger of new york co. a world-wide financial board of the companies said it wo n't seek strategic alternatives to the brokerage industry 's directors

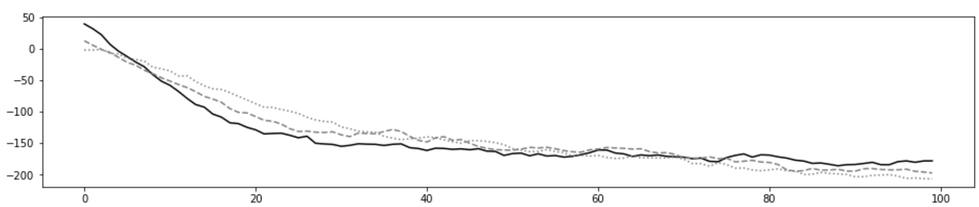

Table 3: Transition of a Markov chain initialized from $p_0(z)$ towards $\tilde{p}_\alpha(z)$. *Top:* Trajectory in the PTB data-space. Each panel contains a sample for $K_0' \in \{0, 40, 100\}$. *Bottom:* Energy profile.

4.4 Anomaly detection

We evaluate our model through the lens of anomaly detection. If the generator and EBM are well learned, then the posterior $p_\theta(z|x)$ would form a discriminative latent space that has separated probability density for normal and anomalous data, respectively. Samples from such latent space can then be used as discriminative features to detect anomalies. We perform posterior sampling on the learned model to obtain the latent samples, and use the unnormalized log-posterior $\log p_\theta(x, z)$ as our decision function.

Following the protocol as in [31, 62], we make each digit class an anomaly and consider the remaining 9 digits as normal examples. Our model is trained with only normal data and tested with both normal and anomalous data. We compare with the BiGAN-based anomaly detection [62], MEG [31] and VAE using area under the precision-recall curve (AUPRC) as in [62]. Table 4 shows the results.

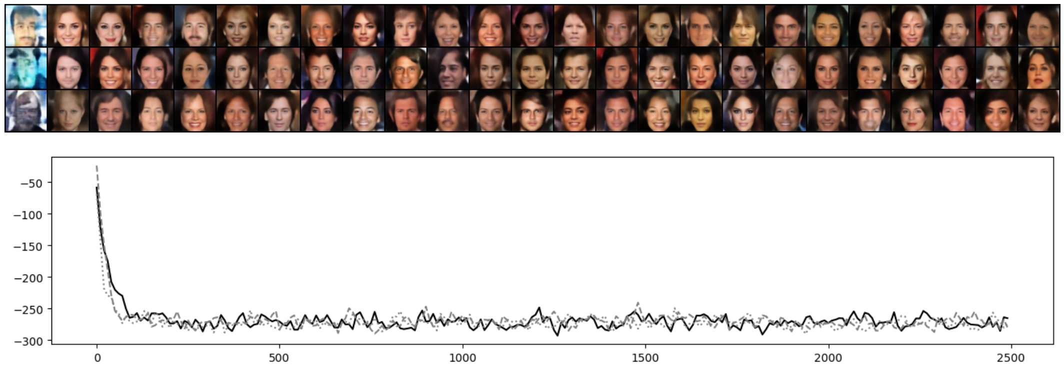

Figure 4: Transition of Markov chains initialized from $p_0(z)$ towards $\tilde{p}_\alpha(z)$ for $K_0' = 2500$ steps. *Top:* Trajectory in the CelebA data-space for every 100 steps. *Bottom:* Energy profile over time.

Heldout Digit	1	4	5	7	9
VAE	0.063	0.337	0.325	0.148	0.104
MEG	0.281 $\pm$ 0.035	0.401 $\pm$0.061	0.402 $\pm$ 0.062	0.290 $\pm$ 0.040	0.342 $\pm$ 0.034
BiGAN-σ	0.287 $\pm$ 0.023	0.443 $\pm$ 0.029	0.514 $\pm$ 0.029	0.347 $\pm$ 0.017	0.307 $\pm$ 0.028
Ours	**0.336 $\pm$ 0.008**	**0.630 $\pm$ 0.017**	**0.619 $\pm$ 0.013**	**0.463 $\pm$ 0.009**	**0.413 $\pm$ 0.010**

Table 4: AUPRC scores for unsupervised anomaly detection on MNIST. Numbers are taken from [31] and results for our model are averaged over last 10 epochs to account for variance.

4.5 Ablation study

We investigate a range of factors that are potentially affecting the model performance with SVHN as an example. The highlighted number in Tables 5, 6, and 7 is the FID score reported in the main text and compared to other baseline models. It is obtained from the model with the architecture and hyperparameters specified in Table 8 and Table 9 which serve as the reference configuration for the ablation study.

Fixed prior. We examine the expressivity endowed with the EBM prior by comparing it to models with a fixed isotropic Gaussian prior. The results are displayed in Table 5. The model with an EBM prior clearly outperforms the model with a fixed Gaussian prior and the same generator as the reference model. The fixed Gaussian models exhibit an enhancement in performance as the generator complexity increases. They however still have an inferior performance compared to the model with an EBM prior even when the fixed Gaussian prior model has a generator with four times more parameters than that of the reference model.

Model	FID
Latent EBM Prior	**29.44**
Fixed Gaussian	
same generator	43.39
generator with 2 times as many parameters	41.10
generator with 4 times as many parameters	39.50

Table 5: Comparison of the models with a latent EBM prior versus a fixed Gaussian prior. The highlighted number is the reported FID for SVHN and compared to other baseline models in the main text.

MCMC steps. We also study how the number of short run MCMC steps for prior inference (K_0) and posterior inference (K_1). The left panel of Table 6 shows the results for K_0 and the right panel for K_1. As the number of MCMC steps increases, we observe improved quality of synthesis in terms of FID.

Steps	FID
$K_0 = 40$	31.49
$K_0 = 60$	**29.44**
$K_0 = 80$	28.32

Steps	FID
$K_1 = 20$	**29.44**
$K_1 = 40$	27.26
$K_1 = 60$	26.13

Table 6: Influence of the number of prior and posterior short run steps K_0 (left) and K_1 (right). The highlighted number is the reported FID for SVHN and compared to other baseline models in the main text.

Prior EBM and generator complexity. Table 7 displays the FID scores as a function of the number of hidden features of the prior EBM (nef) and the factor of the number of channels of the generator (ngf, also see Table 9). In general, enhanced model complexity leads to improved generation.

nef		50	100	200
	32	32.25	31.98	30.78
ngf	64	30.91	30.56	**29.44**
	128	29.12	27.24	26.95

Table 7: Influence of prior and generator complexity. The highlighted number is the reported FID for SVHN and compared to other baseline models in the main text. **nef** indicates the number of hidden features of the prior EBM and **ngf** denotes the factor of the number of channels of the generator (also see Table 9).

4.6 Computational cost

We note that our method involving MCMC sampling is more computationally costly compared to those with amortized inference such as VAE which however bears on issues like inaccurate inference and the mismatch between the prior and aggregate posterior. Several works involve MCMC sampling attempting to improve VAE by either enhancing the posterior inference (SA-VAE [27]) or constructing a flexible prior with rejection sampling (LARS [3]) within the original VAE framework. In contrast, we adopt a maximum likelihood learning approach with short run MCMC sampling and follow the philosophy of empirical Bayes. Our approach trades feasible computational cost for expressive prior and simple and accurate inference. Consider SVHN as an example. Training our model on a single NVIDIA 1080Ti needs approximately 6 hours to converge and VAE training needs 1.5 hours. Thus our method is 4 times slower. Bearing with the feasible cost our method leads to performance improving over strong baselines such as 2sVAE [11] and RAE [16] on image and text modeling and anomaly detection.

We have also explored avenues to improve training speed and found that a PyTorch extension, NVIDIA Apex [2], is able to improve the training time of our model by a factor of 2.5. We test our method with Apex training on a larger scale dataset, CelebA $(128 \times 128 \times 3)$. The learned model is able to synthesize examples with high fidelity (see Figure 1 for examples).

5 Conclusion

This paper proposes a generalization of the generator model, where the latent vector follows a latent space EBM, which is a refinement or correction of the independent Gaussian or uniform noise prior in the original generator model. We adopt a simple maximum likelihood framework for learning, and develop a practical modification of the maximum likelihood learning algorithm based on short-run MCMC sampling from the prior and posterior distributions of the latent vector. We also provide a theoretical underpinning of the resulting algorithm as a perturbation of the maximum likelihood learning in terms of objective function and estimating equation. Our method combines the best of both top-down generative model and undirected EBM.

[2]https://github.com/NVIDIA/apex

EBM has many applications, however, its soundness and its power are limited by the difficulty with MCMC sampling. By moving from data space to latent space, MCMC-based learning of EBM becomes sound and feasible, and we may release the power of EBM in the latent space for many applications.

6 Appendix A: Theoretical derivations

In this section, we shall derive most of the equations in the main text. We take a step by step approach, starting from simple identities or results, and gradually reaching the main results. Our derivations are unconventional, but they pertain more to our model and learning method.

6.1 A simple identity

Let $x \sim p_\theta(x)$. A useful identity is

$$\mathrm{E}_\theta[\nabla_\theta \log p_\theta(x)] = 0, \tag{18}$$

where E_θ (or E_{p_θ}) is the expectation with respect to p_θ.

The proof is one liner:

$$\mathrm{E}_\theta[\nabla_\theta \log p_\theta(x)] = \int [\nabla_\theta \log p_\theta(x)] p_\theta(x) dx = \int \nabla_\theta p_\theta(x) dx = \nabla_\theta \int p_\theta(x) dx = \nabla_\theta 1 = 0. \tag{19}$$

The above identity has generalized versions, such as the one underlying the policy gradient [52], $\nabla_\theta \mathrm{E}_\theta[R(x)] = \mathrm{E}_\theta[R(x)\nabla_\theta \log p_\theta(x)]$. By letting $R(x) = 1$, we get (18).

6.2 Maximum likelihood estimating equation

The simple identity (18) also underlies the consistency of MLE. Suppose we observe $(x_i, i = 1, ..., n) \sim p_{\theta_{\rm true}}(x)$ independently, where $\theta_{\rm true}$ is the true value of θ. The log-likelihood is

$$L(\theta) = \frac{1}{n} \sum_{i=1}^{n} \log p_\theta(x_i). \tag{20}$$

The maximum likelihood estimating equation is

$$L'(\theta) = \frac{1}{n} \sum_{i=1}^{n} \nabla_\theta \log p_\theta(x_i) = 0. \tag{21}$$

According to the law of large number, as $n \to \infty$, the above estimating equation converges to

$$\mathrm{E}_{\theta_{\rm true}}[\nabla_\theta \log p_\theta(x)] = 0, \tag{22}$$

where θ is the unknown value to be solved, while $\theta_{\rm true}$ is fixed. According to the simple identity (18), $\theta = \theta_{\rm true}$ is the solution to the above estimating equation (22), no matter what $\theta_{\rm true}$ is. Thus with regularity conditions, such as identifiability of the model, the MLE converges to $\theta_{\rm true}$ in probability.

The optimality of the maximum likelihood estimating equation among all the asymptotically unbiased estimating equations can be established based on a further generalization of the simple identity (18).

We shall justify our learning method with short run MCMC in terms of an estimating equation, which is a perturbation of the maximum likelihood estimating equation.

6.3 MLE learning gradient for θ

Recall that $p_\theta(x, z) = p_\alpha(z) p_\beta(x|z)$, where $\theta = \{\alpha, \beta\}$. The learning gradient for an observation x is as follows:

$$\nabla_\theta \log p_\theta(x) = \mathrm{E}_{p_\theta(z|x)} \left[\nabla_\theta \log p_\theta(x, z)\right] = \mathrm{E}_{p_\theta(z|x)} \left[\nabla_\theta (\log p_\alpha(z) + \log p_\beta(x|z))\right]. \tag{23}$$

The above identity is a simple consequence of the simple identity (18).

$$
\begin{aligned}
\mathrm{E}_{p_\theta(z|x)}\left[\nabla_\theta \log p_\theta(x,z)\right] &= \mathrm{E}_{p_\theta(z|x)}\left[\nabla_\theta \log p_\theta(z|x) + \nabla_\theta \log p_\theta(x)\right] && (24)\\
&= \mathrm{E}_{p_\theta(z|x)}\left[\nabla_\theta \log p_\theta(z|x)\right] + \mathrm{E}_{p_\theta(z|x)}\left[\nabla_\theta \log p_\theta(x)\right] && (25)\\
&= 0 + \nabla_\theta \log p_\theta(x), && (26)
\end{aligned}
$$

because of the fact that $\mathrm{E}_{p_\theta(z|x)}\left[\nabla_\theta \log p_\theta(z|x)\right] = 0$ according to the simple identity (18), while $\mathrm{E}_{p_\theta(z|x)}\left[\nabla_\theta \log p_\theta(x)\right] = \nabla_\theta \log p_\theta(x)$ because what is inside the expectation only depends on x, but does not depend on z.

The above identity (23) is related to the EM algorithm [14], where x is the observed data, z is the missing data, and $\log p_\theta(x,z)$ is the complete-data log-likelihood.

6.4 MLE learning gradient for α

For the prior model $p_\alpha(z) = \frac{1}{Z(\alpha)}\exp(f_\alpha(z))p_0(z)$, we have $\log p_\alpha(z) = f_\alpha(z) - \log Z(\alpha) + \log p_0(z)$. Applying the simple identity (18), we have

$$
\mathrm{E}_\alpha[\nabla_\alpha \log p_\alpha(z)] = \mathrm{E}_\alpha[\nabla_\alpha f_\alpha(z) - \nabla_\alpha \log Z(\alpha)] = \mathrm{E}_\alpha[\nabla_\alpha f_\alpha(z)] - \nabla_\alpha \log Z(\alpha) = 0. \tag{27}
$$

Thus

$$
\nabla_\alpha \log Z(\alpha) = \mathrm{E}_\alpha[\nabla_\alpha f_\alpha(z)]. \tag{28}
$$

Hence the derivative of the log-likelihood is

$$
\nabla_\alpha \log p_\alpha(x) = \nabla_\alpha f_\alpha(z) - \nabla_\alpha \log Z(\alpha) = \nabla_\alpha f_\alpha(z) - \mathrm{E}_\alpha[\nabla_\alpha f_\alpha(z)]. \tag{29}
$$

According to equation (23) in the previous subsection, the learning gradient for α is

$$
\begin{aligned}
\nabla_\alpha \log p_\theta(x) &= \mathrm{E}_{p_\theta(z|x)}\left[\nabla_\alpha \log p_\alpha(z)\right] && (30)\\
&= \mathrm{E}_{p_\theta(z|x)}[\nabla_\alpha f_\alpha(z) - \mathrm{E}_{p_\alpha(z)}[\nabla_\alpha f_\alpha(z))]] && (31)\\
&= \mathrm{E}_{p_\theta(z|x)}[\nabla_\alpha f_\alpha(z)] - \mathrm{E}_{p_\alpha(z)}[\nabla_\alpha f_\alpha(z)]. && (32)
\end{aligned}
$$

6.5 Re-deriving simple identity in terms of D_{KL}

We shall provide a theoretical understanding of the learning method with short run MCMC in terms of Kullback-Leibler divergences. We start from some simple results.

The simple identity (18) also follows from Kullback-Leibler divergence. Consider

$$
D(\theta) = D_{KL}(p_{\theta_*}(x) \| p_\theta(x)), \tag{33}
$$

as a function of θ with θ_* fixed. Suppose the model p_θ is identifiable, then $D(\theta)$ achieves its minimum 0 at $\theta = \theta_*$, thus $D'(\theta_*) = 0$. Meanwhile,

$$
D'(\theta) = -\mathrm{E}_{\theta_*}[\nabla_\theta \log p_\theta(x)]. \tag{34}
$$

Thus

$$
\mathrm{E}_{\theta_*}[\nabla_\theta \log p_{\theta_*}(x)] = 0. \tag{35}
$$

Since θ_* is arbitrary in the above derivation, we can replace it by a generic θ, i.e.,

$$
\mathrm{E}_\theta[\nabla_\theta \log p_\theta(x)] = 0, \tag{36}
$$

which is the simple identity (18).

As a notational convention, for a function $f(\theta)$, we write $f'(\theta_*) = \nabla_\theta f(\theta_*)$, i.e., the derivative of $f(\theta)$ at θ_*.

6.6 Re-deriving MLE learning gradient in terms of perturbation by D_{KL} terms

We now re-derive MLE learning gradient in terms of perturbation of log-likelihood by Kullback-Leibler divergence terms. Then the learning method with short run MCMC can be easily understood.

At iteration t, fixing θ_t, we want to calculate the gradient of the log-likelihood function for an observation x, $\log p_\theta(x)$, at $\theta = \theta_t$. Consider the following perturbation of the log-likelihood

$$l(\theta) = \log p_\theta(x) - D_{KL}(p_{\theta_t}(z|x) \| p_\theta(z|x)) + D_{KL}(p_{\alpha_t}(z) \| p_\alpha(z)). \tag{37}$$

In the above, as a function of θ, with θ_t fixed, $D_{KL}(p_{\theta_t}(z|x) \| p_\theta(z|x))$ is minimized at $\theta = \theta_t$, thus its derivative at θ_t is 0. As a function of α, with α_t fixed, $D_{KL}(p_{\alpha_t}(z) \| p_\alpha(z))$ is minimized at $\alpha = \alpha_t$, thus its derivative at α_t is 0. Thus

$$\nabla_\theta \log p_{\theta_t}(x) = l'(\theta_t). \tag{38}$$

We now unpack $l(\theta)$ so that we can obtain its derivative at θ_t.

$$\begin{aligned}
l(\theta) &= \log p_\theta(x) + \mathrm{E}_{p_{\theta_t}(z|x)}[\log p_\theta(z|x)] - \mathrm{E}_{p_{\alpha_t}(z)}[\log p_\alpha(z)] + c && (39)\\
&= \mathrm{E}_{p_{\theta_t}(z|x)}[\log p_\theta(x, z)] - \mathrm{E}_{p_{\alpha_t}(z)}[\log p_\alpha(z)] + c && (40)\\
&= \mathrm{E}_{p_{\theta_t}(z|x)}[\log p_\alpha(z) + \log p_\beta(x|z)] - \mathrm{E}_{p_{\alpha_t}(z)}[\log p_\alpha(z)] + c && (41)\\
&= \mathrm{E}_{p_{\theta_t}(z|x)}[\log p_\alpha(z)] - \mathrm{E}_{p_{\alpha_t}(z)}[\log p_\alpha(z)] + \mathrm{E}_{p_{\theta_t}(z|x)}[\log p_\beta(x|z)] + c && (42)\\
&= \mathrm{E}_{p_{\theta_t}(z|x)}[f_\alpha(z)] - \mathrm{E}_{p_{\alpha_t}(z)}[f_\alpha(z)] + \mathrm{E}_{p_{\theta_t}(z|x)}[\log p_\beta(x|z)] + c + c', && (43)
\end{aligned}$$

where $\log Z(\alpha)$ term gets canceled,

$$\begin{aligned}
c &= -\mathrm{E}_{p_{\theta_t}(z|x)}[\log p_{\theta_t}(z|x)] + \mathrm{E}_{p_{\alpha_t}(z)}[\log p_{\alpha_t}(z)], && (44)\\
c' &= \mathrm{E}_{p_{\theta_t}(z|x)}[\log p_0(z)] - \mathrm{E}_{p_{\alpha_t}(z)}[\log p_0(z)], && (45)
\end{aligned}$$

do not depend on θ. c consists of two entropy terms. Now taking derivative at θ_t, we have

$$\begin{aligned}
\delta_{\alpha_t}(x) &= \nabla_\alpha l(\theta_t) = \mathrm{E}_{p_{\theta_t}(z|x)}[\nabla_\alpha f_{\alpha_t}(z)] - \mathrm{E}_{p_{\alpha_t}(z)}[\nabla_\alpha f_{\alpha_t}(z)], && (46)\\
\delta_{\beta_t}(x) &= \nabla_\beta l(\theta_t) = \mathrm{E}_{p_{\theta_t}(z|x)}[\nabla_\beta \log p_{\beta_t}(x|z)]. && (47)
\end{aligned}$$

Averaging over the observed examples $\{x_i, i = 1, ..., n\}$ leads to MLE learning gradient.

In the above, we calculate the gradient of $\log p_\theta(x)$ at θ_t. Since θ_t is arbitrary in the above derivation, if we replace θ_t by a generic θ, we get the gradient of $\log p_\theta(x)$ at a generic θ, i.e.,

$$\begin{aligned}
\delta_\alpha(x) &= \nabla_\alpha \log p_\theta(x) = \mathrm{E}_{p_\theta(z|x)}[\nabla_\alpha f_\alpha(z)] - \mathrm{E}_{p_\alpha(z)}[\nabla_\alpha f_\alpha(z)], && (48)\\
\delta_\beta(x) &= \nabla_\beta \log p_\theta(x) = \mathrm{E}_{p_\theta(z|x)}[\nabla_\beta \log p_\beta(x|z)]. && (49)
\end{aligned}$$

The above calculations are related to the EM algorithm and the learning of energy-based model.

In EM algorithm [14], the complete-data log-likelihood Q serves as a surrogate for the observed-data log-likelihood $\log p_\theta(x)$, where

$$Q(\theta|\theta_t) = \log p_\theta(x) - D_{KL}(p_{\theta_t}(z|x) \| p_\theta(z|x)), \tag{50}$$

and $\theta_{t+1} = \arg\max_\theta Q(\theta|\theta_t)$, where $Q(\theta|\theta_t)$ is a lower-bound of $\log p_\theta(x)$ or minorizes the latter. $Q(\theta|\theta_t)$ and $\log p_\theta(x)$ touch each other at θ_t, and they are co-tangent at θ_t. Thus the derivative of $\log p_\theta(x)$ at θ_t is the same as the derivative of $Q(\theta|\theta_t)$ at $\theta = \theta_t$.

In EBM, $D_{KL}(p_{\alpha_t}(z) \| p_\alpha(z))$ serves to cancel $\log Z(\alpha)$ term in the EBM prior, and is related to the second divergence term in contrastive divergence.

6.7 Maximum likelihood estimating equation for $\theta = (\alpha, \beta)$

The MLE estimating equation is

$$\frac{1}{n} \sum_{i=1}^{n} \nabla_\theta \log p_\theta(x_i) = 0. \tag{51}$$

Based on (48) and (49), the estimating equation is

$$\frac{1}{n}\sum_{i=1}^{n}\delta_\alpha(x_i) = \frac{1}{n}\sum_{i=1}^{n}\mathrm{E}_{p_\theta(z_i|x_i)}[\nabla_\alpha f_\alpha(z_i)] - \mathrm{E}_{p_\alpha(z)}[\nabla_\alpha f_\alpha(z)] = 0, \tag{52}$$

$$\frac{1}{n}\sum_{i=1}^{n}\delta_\beta(x_i) = \frac{1}{n}\sum_{i=1}^{n}\mathrm{E}_{p_\theta(z_i|x_i)}[\nabla_\beta \log p_\beta(x_i|z_i)] = 0. \tag{53}$$

6.8 Learning with short run MCMC as perturbation of log-likelihood

Based on the above derivations, we can see that learning with short run MCMC is also a perturbation of log-likelihood, except that we replace $p_{\theta_t}(z|x)$ by $\tilde{p}_{\theta_t}(z|x)$, and replace $p_{\alpha_t}(z)$ by $\tilde{p}_{\alpha_t}(z)$, where $\tilde{p}_{\theta_t}(z|x)$ and $\tilde{p}_{\alpha_t}(z)$ are produced by short run MCMC.

At iteration t, fixing θ_t, the updating rule based on short run MCMC follows the gradient of the following function, which is a perturbation of log-likelihood for the observation x,

$$\tilde{l}(\theta) = \log p_\theta(x) - D_{KL}(\tilde{p}_{\theta_t}(z|x)\|p_\theta(z|x)) + D_{KL}(\tilde{p}_{\alpha_t}(z)\|p_\alpha(z)). \tag{54}$$

The above is a function of θ, while θ_t is fixed. In the main text, we write $\tilde{l}(\theta) = \log \tilde{p}_\theta(x)$ to emphasize that $\tilde{l}(\theta)$ is a perturbation of $\log p_\theta(x)$. $\tilde{p}_\theta(x)$ is a convenient notation. It is not necessarily a normalized probability density function.

In full parallel to the above subsection, we have

$$\begin{aligned}
\tilde{l}(\theta) &= \log p_\theta(x) + \mathrm{E}_{\tilde{p}_{\theta_t}(z|x)}[\log p_\theta(z|x)] - \mathrm{E}_{\tilde{p}_{\alpha_t}(z)}[\log p_\alpha(z)] + c &(55)\\
&= \mathrm{E}_{\tilde{p}_{\theta_t}(z|x)}[\log p_\theta(x,z)] - \mathrm{E}_{\tilde{p}_{\alpha_t}(z)}[\log p_\alpha(z)] + c &(56)\\
&= \mathrm{E}_{\tilde{p}_{\theta_t}(z|x)}[\log p_\alpha(z) + \log p_\beta(x|z)] - \mathrm{E}_{\tilde{p}_{\alpha_t}(z)}[\log p_\alpha(z)] + c &(57)\\
&= \mathrm{E}_{\tilde{p}_{\theta_t}(z|x)}[\log p_\alpha(z)] - \mathrm{E}_{\tilde{p}_{\alpha_t}(z)}[\log p_\alpha(z)] + \mathrm{E}_{\tilde{p}_{\theta_t}(z|x)}[\log p_\beta(x|z)] + c &(58)\\
&= \mathrm{E}_{\tilde{p}_{\theta_t}(z|x)}[f_\alpha(z)] - \mathrm{E}_{\tilde{p}_{\alpha_t}(z)}[f_\alpha(z)] + \mathrm{E}_{\tilde{p}_{\theta_t}(z|x)}[\log p_\beta(x|z)] + c + c', &(59)
\end{aligned}$$

where $\log Z(\alpha)$ term gets canceled,

$$c = -\mathrm{E}_{\tilde{p}_{\theta_t}(z|x)}[\log \tilde{p}_{\theta_t}(z|x)] + \mathrm{E}_{\tilde{p}_{\alpha_t}(z)}[\log \tilde{p}_{\alpha_t}(z)], \tag{60}$$

$$c' = \mathrm{E}_{\tilde{p}_{\theta_t}(z|x)}[\log p_0(z)] - \mathrm{E}_{\tilde{p}_{\alpha_t}(z)}[\log p_0(z)], \tag{61}$$

do not depend on θ. c consists of two entropy terms. Thus, taking derivative of the function $\tilde{l}(\theta)$ at $\theta = \theta_t$, we have

$$\tilde{\delta}_{\alpha_t}(x) = \nabla_\alpha \tilde{l}(\theta_t) = \mathrm{E}_{\tilde{p}_{\theta_t}(z|x)}[\nabla_\alpha f_{\alpha_t}(z)] - \mathrm{E}_{\tilde{p}_{\alpha_t}(z)}[\nabla_\alpha f_{\alpha_t}(z)], \tag{62}$$

$$\tilde{\delta}_{\beta_t}(x) = \nabla_\beta \tilde{l}(\theta_t) = \mathrm{E}_{\tilde{p}_{\theta_t}(z|x)}[\nabla_\beta \log p_{\beta_t}(x|z)]. \tag{63}$$

Averaging over $\{x_i, i = 1, ..., n\}$, we get the updating rule based on short run MCMC. That is, the learning rule based on short run MCMC follows the gradient of a perturbation of the log-likelihood function where the perturbations consists of two D_{KL} terms.

$D_{KL}(\tilde{p}_{\theta_t}(z|x)\|p_\theta(z|x))$ is related to VAE [29], where $\tilde{p}_{\theta_t}(z|x)$ serves as an inference model, except that we do not learn a separate inference network. $D_{KL}(\tilde{p}_{\alpha_t}(z)\|p_\alpha(z))$ is related to contrastive divergence [23], except that $\tilde{p}_{\alpha_t}(z)$ is initialized from the Gaussian white noise $p_0(z)$, instead of the data distribution of observed examples.

$D_{KL}(\tilde{p}_{\theta_t}(z|x)\|p_\theta(z|x))$ and $D_{KL}(\tilde{p}_{\alpha_t}(z)\|p_\alpha(z))$ cause the bias relative to MLE learning, which is impractical because we cannot do exact sampling with MCMC.

However, the bias may not be all that bad. In learning β, $D_{KL}(\tilde{p}_{\theta_t}(z|x)\|p_\theta(z|x))$ may force the model to be biased towards the approximate short run posterior $\tilde{p}_{\theta_t}(z|x)$, so that the short run posterior is close to the true posterior. In learning α, the update based on $\mathrm{E}_{\tilde{p}_\theta(z|x)}[\nabla_\alpha f_\alpha(z)] - \mathrm{E}_{\tilde{p}_\alpha(z)}[\nabla_\alpha f_\alpha(z)]$ may force the short run prior $\tilde{p}_\alpha(z)$ to match the short run posterior $\tilde{p}_\theta(z|x)$. We shall investigate this issue in future work.

6.9 Perturbation of maximum likelihood estimating equation

The fixed point of the learning algorithm based on short run MCMC is where the updating is 0, i.e.,

$$\frac{1}{n}\sum_{i=1}^{n}\tilde{\delta}_{\alpha}(x_i) = \frac{1}{n}\sum_{i=1}^{n}\mathrm{E}_{\tilde{p}_{\theta}(z_i|x_i)}[\nabla_{\alpha}f_{\alpha}(z_i)] - \mathrm{E}_{\tilde{p}_{\alpha}(z)}[\nabla_{\alpha}f_{\alpha}(z)] = 0, \tag{64}$$

$$\frac{1}{n}\sum_{i=1}^{n}\tilde{\delta}_{\beta}(x_i) = \frac{1}{n}\sum_{i=1}^{n}\mathrm{E}_{\tilde{p}_{\theta}(z_i|x_i)}[\nabla_{\beta}\log p_{\beta}(x_i|z_i)] = 0. \tag{65}$$

This is clearly a perturbation of the MLE estimating equation in (52) and (53). The above estimating equation defines an estimator, where the learning algorithm with short run MCMC converges.

6.10 Three D_{KL} terms

We can rewrite the objective function (54) in a more revealing form. Let $(x_i, i = 1, ..., n) \sim p_{\text{data}}(x)$ independently, where $p_{\text{data}}(x)$ is the data distribution. At time step t, with fixed θ_t, learning based on short-run MCMC follows the gradient of

$$\frac{1}{n}\sum_{i=1}^{n}[\log p_{\theta}(x_i) - D_{KL}(\tilde{p}_{\theta_t}(z_i|x_i)\|p_{\theta}(z_i|x_i)) + D_{KL}(\tilde{p}_{\alpha_t}(z)\|p_{\alpha}(z))]. \tag{66}$$

Let us assume n is large enough, so that the average is practically the expectation with respect to p_{data}. Then MLE maximizes $\frac{1}{n}\sum_{i=1}^{n}\log p_{\theta}(x_i) \doteq \mathrm{E}_{p_{\text{data}}(x)}[\log p_{\theta}(x)]$, which is equivalent to minimizing $D_{KL}(p_{\text{data}}(x)\|p_{\theta}(x))$. The learning with short-run MCMC follows the gradient that minimizes

$$D_{KL}(p_{\text{data}}(x)\|p_{\theta}(x)) + D_{KL}(\tilde{p}_{\theta_t}(z|x)\|p_{\theta}(z|x)) - D_{KL}(\tilde{p}_{\alpha_t}(z)\|p_{\alpha}(z)), \tag{67}$$

where, with some abuse of notation, we now define

$$D_{KL}(\tilde{p}_{\theta_t}(z|x)\|p_{\theta}(z|x)) = \mathrm{E}_{p_{\text{data}}(x)}\mathrm{E}_{\tilde{p}_{\theta_t}(z|x)}\left[\log\frac{\tilde{p}_{\theta_t}(z|x)}{p_{\theta}(z|x)}\right], \tag{68}$$

where we also average over $x \sim p_{\text{data}}(x)$, instead fixing x as before.

The objective (67) is clearly a perturbation of the MLE based on the first D_{KL} in (67). The signs in front of the remaining two D_{KL} perturbations also become clear. The sign in front of $D_{KL}(\tilde{p}_{\theta_t}(z|x)\|p_{\theta}(z|x))$ is positive because

$$D_{KL}(p_{\text{data}}(x)\|p_{\theta}(x)) + D_{KL}(\tilde{p}_{\theta_t}(z|x)\|p_{\theta}(z|x)) = D_{KL}(p_{\text{data}}(x)\tilde{p}_{\theta_t}(z|x)\|p_{\alpha}(z)p_{\beta}(x|z)), \tag{69}$$

and the D_{KL} on the right hand side between the joint distributions of (x, z) are more tractable than the first D_{KL} on the left hand side, which is for MLE. This underlies EM and VAE. Now subtracting the third D_{KL}, we have the following special form of contrastive divergence

$$D_{KL}(p_{\text{data}}(x)\tilde{p}_{\theta_t}(z|x)\|p_{\alpha}(z)p_{\beta}(x|z)) - D_{KL}(\tilde{p}_{\alpha_t}(z)\|p_{\alpha}(z)), \tag{70}$$

where the negative sign in front of $D_{KL}(\tilde{p}_{\alpha_t}(z)\|p_{\alpha}(z))$ is to cancel the intractable $\log Z(\alpha)$ term.

The above contrastive divergence also has an adversarial interpretation, where $p_{\alpha}(z)$ or α is updated, so that $p_{\alpha}(z)p_{\beta}(x|z)$ gets closer to $p_{\text{data}}(x)\tilde{p}_{\theta_t}(z|x)$, while getting away from $\tilde{p}_{\alpha_t}(z)$, i.e., p_{α} seeks to criticize the samples from $\tilde{p}_{\alpha_t}(z)$ by comparing them to the posterior samples of z inferred from the real data.

As mentioned in the main text, we can also exponential tilt $p_0(x, z) = p_0(z)p_{\beta}(x|z)$ to $p_{\theta}(x, z) = \frac{1}{Z(\theta)}\exp(f_{\alpha}(x, z))p_0(x, z)$, or equivalently, exponentially tilt $p_0(z, \epsilon) = p_0(z)p(\epsilon)$. The above derivations can be easily adapted to such a model, which we choose not to explore due to the complexity of EBM in the data space.

7 Appendix B: Experiments

7.1 Experiment details

Data. Image datasets include SVHN [41] ($32 \times 32 \times 3$), CIFAR-10 [30] ($32 \times 32 \times 3$), and CelebA [36] ($64 \times 64 \times 3$). We use the full training split of SVHN ($73,257$) and CIFAR-10 ($50,000$)

and take $40,000$ examples of CelebA as training data following [44]. The training images are resized and scaled to $[-1, 1]$. Text datasets include PTB [40], Yahoo [61], and SNLI [6], following recent work on text generative modeling with latent variables [27, 63, 35].

Model architectures. The architecture of the EBM, $f_\alpha(z)$, is displayed in Table 9. For text data, the dimensionality of z is set to 32. The generator architectures for the image data are also shown in Table 9. The generators for the text data are implemented with a one-layer unidirectional LSTM [25] and Table 10 lists the number of word embeddings and hidden units of the generators for each dataset.

Short run dynamics. The hyperparameters for the short run dynamics are depicted in Table 8 where K_0 and K_1 denote the number of prior and posterior sampling steps with step sizes s_0 and s_1, respectively. These are identical across models and data modalities, except for the model for CIFAR-10 which is using $K_1 = 40$ steps.

Short Run Dynamics Hyperparameters	
Hyperparameter	Value
K_0	60
s_0	0.4
K_1	20
s_1	0.1

Table 8: Hyperparameters for short run dynamics.

Optimization. The parameters for the EBM and image generators are initialized with Xavier normal [17] and those for the text generators are initialized from a uniform distribution, Unif$(-0.1, 0.1)$, following [27, 35]. Adam [28] is adopted for all model optimization. The models are trained until convergence (taking approximately $70,000$ and $40,000$ parameter updates for image and text models, respectively).

	SNLI	PTB	Yahoo
Word Embedding Size	256	128	512
Hidden Size of Generator	256	512	1024

Table 10: The sizes of word embeddings and hidden units of the generators for SNLI, PTB, and Yahoo.

Acknowledgments

The work is supported by DARPA XAI project N66001-17-2-4029; ARO project W911NF1810296; ONR MURI project N00014-16-1-2007; and XSEDE grant ASC170063. We thank the NVIDIA cooperation for the donation of 2 Titan V GPUs.

EBM Model		
Layers	In-Out Size	Stride
Input: z	100	
Linear, LReLU	200	-
Linear, LReLU	200	-
Linear	1	-
Generator Model for SVHN, ngf = 64		
Input: x	1x1x100	
4x4 convT(ngf x 8), LReLU	4x4x(ngf x 8)	1
4x4 convT(ngf x 4), LReLU	8x8x(ngf x 4)	2
4x4 convT(ngf x 2), LReLU	16x16x(ngf x 2)	2
4x4 convT(3), Tanh	32x32x3	2
Generator Model for CIFAR-10, ngf = 128		
Input: x	1x1x128	
8x8 convT(ngf x 8), LReLU	8x8x(ngf x 8)	1
4x4 convT(ngf x 4), LReLU	16x16x(ngf x 4)	2
4x4 convT(ngf x 2), LReLU	32x32x(ngf x 2)	2
3x3 convT(3), Tanh	32x32x3	1
Generator Model for CelebA, ngf = 128		
Input: x	1x1x100	
4x4 convT(ngf x 8), LReLU	4x4x(ngf x 8)	1
4x4 convT(ngf x 4), LReLU	8x8x(ngf x 4)	2
4x4 convT(ngf x 2), LReLU	16x16x(ngf x 2)	2
4x4 convT(ngf x 1), LReLU	32x32x(ngf x 1)	2
4x4 convT(3), Tanh	64x64x3	2

Table 9: EBM model architectures for all image and text datasets and generator model architectures for SVHN $(32 \times 32 \times 3)$, CIFAR-10 $(32 \times 32 \times 3)$, and CelebA $(64 \times 64 \times 3)$. convT(n) indicates a transposed convolutional operation with n output feature maps. LReLU indicates the Leaky-ReLU activation function. The leak factor for LReLU is 0.2 in EBM and 0.1 in Generator.

References

[1] David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski. A learning algorithm for boltzmann machines. *Cognitive Science*, 9(1):147–169, 1985.

[2] Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 214–223, 2017.

[3] Matthias Bauer and Andriy Mnih. Resampled priors for variational autoencoders. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 66–75, 2019.

[4] Yoshua Bengio, Grégoire Mesnil, Yann Dauphin, and Salah Rifai. Better mixing via deep representations. In *International conference on machine learning*, pages 552–560, 2013.

[5] Piotr Bojanowski, Armand Joulin, David Lopez-Paz, and Arthur Szlam. Optimizing the latent space of generative networks. *arXiv preprint arXiv:1707.05776*, 2017.

[6] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, Sept. 2015. Association for Computational Linguistics.

[7] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany, Aug. 2016. Association for Computational Linguistics.

[8] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[9] Ondřej Cífka, Aliaksei Severyn, Enrique Alfonseca, and Katja Filippova. Eval all, trust a few, do wrong to none: Comparing sentence generation models. *arXiv preprint arXiv:1804.07972*, 2018.

[10] Thomas M. Cover and Joy A. Thomas. *Elements of information theory (2. ed.)*. Wiley, 2006.

[11] Bin Dai and David Wipf. Diagnosing and enhancing vae models. *arXiv preprint arXiv:1903.05789*, 2019.

[12] Bin Dai and David Wipf. Diagnosing and enhancing vae models. In *International Conference on Learning Representations*, 2019.

[13] Zihang Dai, Amjad Almahairi, Philip Bachman, Eduard H. Hovy, and Aaron C. Courville. Calibrating energy-based generative adversarial networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[14] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[15] Yilun Du and Igor Mordatch. Implicit generation and generalization in energy-based models. *CoRR*, abs/1903.08689, 2019.

[16] Partha Ghosh, Mehdi S. M. Sajjadi, Antonio Vergari, Michael Black, and Bernhard Scholkopf. From variational to deterministic autoencoders. In *International Conference on Learning Representations*, 2020.

[17] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[18] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680, 2014.

[19] Cheng-En Guo, Song-Chun Zhu, and Ying Nian Wu. Modeling visual patterns by integrating descriptive and generative methods. *International Journal of Computer Vision*, 53(1):5–29, 2003.

[20] Tian Han, Yang Lu, Song-Chun Zhu, and Ying Nian Wu. Alternating back-propagation for generator network. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 1976–1984, 2017.

[21] Tian Han, Erik Nijkamp, Xiaolin Fang, Mitch Hill, Song-Chun Zhu, and Ying Nian Wu. Divergence triangle for joint training of generator model, energy-based model, and inferential model. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 8670–8679, 2019.

[22] Tian Han, Erik Nijkamp, Linqi Zhou, Bo Pang, Song-Chun Zhu, and Ying Nian Wu. Joint training of variational auto-encoder and latent energy-based model. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

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

[24] Geoffrey E Hinton, Peter Dayan, Brendan J Frey, and Radford M Neal. The" wake-sleep" algorithm for unsupervised neural networks. *Science*, 268(5214):1158–1161, 1995.

[25] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[26] Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent component analysis*. John Wiley & Sons, 2004.

[27] Yoon Kim, Sam Wiseman, Andrew Miller, David Sontag, and Alexander Rush. Semi-amortized variational autoencoders. In *International Conference on Machine Learning*, pages 2678–2687, 2018.

[28] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[29] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[30] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research).

[31] Rithesh Kumar, Anirudh Goyal, Aaron C. Courville, and Yoshua Bengio. Maximum entropy generators for energy-based models. *CoRR*, abs/1901.08508, 2019.

[32] Paul Langevin. *On the theory of Brownian motion*. 1908.

[33] Justin Lazarow, Long Jin, and Zhuowen Tu. Introspective neural networks for generative modeling. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2793–2802, 2017.

[34] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[35] Bohan Li, Junxian He, Graham Neubig, Taylor Berg-Kirkpatrick, and Yiming Yang. A surprisingly effective fix for deep latent variable modeling of text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3603–3614, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.

[36] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[37] Yang Lu, Song-Chun Zhu, and Ying Nian Wu. Learning FRAME models using CNN filters. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 1902–1910, 2016.

[38] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are gans created equal? a large-scale study. *arXiv preprint arXiv:1711.10337*, 2017.

[39] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[40] Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Comput. Linguist.*, 19(2):313–330, June 1993.

[41] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[42] Erik Nijkamp, Mitch Hill, Tian Han, Song-Chun Zhu, and Ying Nian Wu. On the anatomy of MCMC-based maximum likelihood learning of energy-based models. *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

[43] Erik Nijkamp, Mitch Hill, Song-Chun Zhu, and Ying Nian Wu. Learning non-convergent non-persistent short-run MCMC toward energy-based model. *Advances in Neural Information*

Processing Systems 33: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, Canada, 2019.

[44] Erik Nijkamp, Bo Pang, Tian Han, Alex Zhou, Song-Chun Zhu, and Ying Nian Wu. Learning deep generative models with short run inference dynamics. *arXiv preprint arXiv:1912.01909*, 2019.

[45] Erik Nijkamp, Bo Pang, Tian Han, Alex Zhou, Song-Chun Zhu, and Ying Nian Wu. Learning deep generative models with short run inference dynamics. *arXiv preprint arXiv:1912.01909*, 2019.

[46] Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997.

[47] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[48] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 1278–1286, 2014.

[49] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[50] Donald B. Rubin and Dorothy T. Thayer. Em algorithms for ml factor analysis. *Psychometrika*, 47(1):69–76, Mar 1982.

[51] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[52] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

[53] Tijmen Tieleman. Training restricted boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th international conference on Machine learning*, pages 1064–1071, 2008.

[54] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558*, 2017.

[55] Jakub Tomczak and Max Welling. Vae with a vampprior. In *International Conference on Artificial Intelligence and Statistics*, pages 1214–1223, 2018.

[56] Jakub M. Tomczak and Max Welling. VAE with a vampprior. In Amos J. Storkey and Fernando Pérez-Cruz, editors, *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*, volume 84 of *Proceedings of Machine Learning Research*, pages 1214–1223. PMLR, 2018.

[57] Ryan D. Turner, Jane Hung, Eric Frank, Yunus Saatchi, and Jason Yosinski. Metropolis-hastings generative adversarial networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 6345–6353. PMLR, 2019.

[58] Ying Nian Wu, Ruiqi Gao, Tian Han, and Song-Chun Zhu. A tale of three probabilistic families: Discriminative, descriptive, and generative models. *Quarterly of Applied Mathematics*, 77(2):423–465, 2019.

[59] Jianwen Xie, Yang Lu, Song-Chun Zhu, and Ying Nian Wu. A theory of generative convnet. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 2635–2644, 2016.

[60] Jianwen Xie, Song-Chun Zhu, and Ying Nian Wu. Synthesizing dynamic patterns by spatial-temporal generative convnet. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1061–1069, 2017.

[61] Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. Improved variational autoencoders for text modeling using dilated convolutions. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 3881–3890, 2017.

[62] Houssam Zenati, Chuan Sheng Foo, Bruno Lecouat, Gaurav Manek, and Vijay Ramaseshan Chandrasekhar. Efficient gan-based anomaly detection. *arXiv preprint arXiv:1802.06222*, 2018.

[63] Junbo Zhao, Yoon Kim, Kelly Zhang, Alexander Rush, and Yann LeCun. Adversarially regularized autoencoders. In *International Conference on Machine Learning*, pages 5902–5911, 2018.

[64] Song Chun Zhu. Statistical modeling and conceptualization of visual patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(6):691–712, 2003.

[65] Song Chun Zhu and David Mumford. Grade: Gibbs reaction and diffusion equations. In *Computer Vision, 1998. Sixth International Conference on*, pages 847–854, 1998.

[66] Song Chun Zhu, Ying Nian Wu, and David Mumford. Filters, random fields and maximum entropy (FRAME): towards a unified theory for texture modeling. *International Journal of Computer Vision*, 27(2):107–126, 1998.