

# Learning Hierarchical Features with Joint Latent Space Energy-Based Prior

Jiali Cui<sup>1</sup>, Ying Nian Wu<sup>2</sup>, Tian Han<sup>1</sup>

<sup>1</sup>Department of Computer Science, Stevens Institute of Technology

<sup>2</sup>Department of Statistics, University of California, Los Angeles

{jcui7, than6}@stevens.edu, ywu@stat.ucla.edu

## Abstract

*This paper studies the fundamental problem of multi-layer generator models in learning hierarchical representations. The multi-layer generator model that consists of multiple layers of latent variables organized in a top-down architecture tends to learn multiple levels of data abstraction. However, such multi-layer latent variables are typically parameterized to be Gaussian, which can be less informative in capturing complex abstractions, resulting in limited success in hierarchical representation learning. On the other hand, the energy-based (EBM) prior is known to be expressive in capturing the data regularities, but it often lacks the hierarchical structure to capture different levels of hierarchical representations. In this paper, we propose a joint latent space EBM prior model with multi-layer latent variables for effective hierarchical representation learning. We develop a variational joint learning scheme that seamlessly integrates an inference model for efficient inference. Our experiments demonstrate that the proposed joint EBM prior is effective and expressive in capturing hierarchical representations and modelling data distribution.*

## 1. Introduction

In recent years, deep generative models have achieved remarkable success in generating high-quality images [36, 31], texts [40, 13], and videos [28, 35]. However, learning generative models that incorporate hierarchical structures still remains a challenge. Such hierarchical generative models can play a critical role in enabling explainable artificial intelligence, thus representing an important area of ongoing research.

To tackle this challenge, various methods of learning hierarchical representations have been explored [30, 42, 21, 3, 30]. These methods learn generative models with multiple layers of latent variables organized in a top-down ar-

chitecture, which have shown the capability of learning increasingly abstract representations (e.g., general structures, classes) at the top layers while capturing low-level data features (e.g., colors, background) at the bottom layers. In general, one could categorize these methods into two classes: (i) conditional hierarchy and (ii) architectural hierarchy.

The conditional hierarchies [30, 12, 25, 21, 3] employ stacked generative models layered on top of one another and assume conditional Gaussian distributions at different layers, while the architectural hierarchies [42, 20] instead leverage network architecture to place high-level representations at the top layers of latent variables and low-level representations at the bottom layers. However, they typically assume conditional Gaussian distribution or isotropic Gaussian as the prior, which could have limited expressivity [26]. Learning a more expressive and informative prior model for multiple layers of latent variables is thus needed.

The energy-based models (EBMs) [19, 39, 7, 24, 26, 4], on the other hand, are shown to be expressive and proved to be powerful in capturing data regularities. Notably, [26] studies the EBM in the latent space, where the energy function is considered as a correction of the non-informative prior and tilts the non-informative prior to a more expressive prior distribution. The low dimensionality of the latent space makes EBM more effective in capturing regularities in the data. However, prior methods often rely on expensive MCMC methods for posterior sampling of the latent variables, and more importantly, the latent variables are not hierarchically organized, making it difficult to capture data variations at different levels.

In this paper, we propose to combine the strengths of latent space EBM and the multi-layer generator model for effective hierarchical representation learning. In particular, we build the EBM on the latent variables across different layers of the generator model, where the latent variables at different layers are concatenated and modelled jointly. The latent variables at different layers capture different levels of information from the data via the hierarchy of the generative model, and their inter-relations are further tightened up and better captured through EBM in joint latent space.

---

Our project page is available at <https://jcui1224.github.io/hierarchical-representation-ebm-proj/>.

Learning the EBM in the latent space can be challenging since it usually requires MCMC sampling for both the prior and posterior distributions of the latent variables. The prior sampling is efficient with the low dimensionality of the latent space and the lightweight network of the energy function, while the MCMC sampling of posterior distribution requires the backward propagation of generation network, which is usually heavy and thus renders inefficient inference. To ensure efficient inference, we introduce an inference model and develop a joint training scheme to efficiently and effectively learn the latent space EBM and multi-layer generator model. In particular, the inference model aims to approximate the exact posterior of the multi-layer latent variables while also serving to extract levels of data variations through the deep feed-forward inference network to facilitate hierarchical learning.

**Contributions:** 1) We propose the joint EBM prior model for hierarchical generator models with multiple layers of latent variables; 2) We develop a variational training scheme for efficient learning and inference; 3) We provide strong empirical results on hierarchical learning and image modeling, as well as various ablation studies to illustrate the effectiveness of the proposed model.

## 2. Background

In this section, we present the background of multi-layer generator models and the latent space EBM, which will serve as the foundation of the proposed model.

### 2.1. Multi-layer generator models

Let  $\mathbf{x} \in \mathbb{R}^D$  represent the high-dimensional observed examples, and  $\mathbf{z} \in \mathbb{R}^d$  denote the low-dimensional latent variables. The multi-layer generator model [30, 36, 21] consists of multiple layers of latent variables that are organized in a top-down hierarchical structure and modelled to be conditionally dependent on its upper layer, i.e.,  $p_\theta(\mathbf{z}) = \prod_{i=1}^{L-1} p_{\theta_i}(\mathbf{z}_i | \mathbf{z}_{i+1}) p_0(\mathbf{z}_L)$ , where  $p_{\theta_i}(\mathbf{z}_i | \mathbf{z}_{i+1}) \sim \mathcal{N}(\mu_{\theta_i}(\mathbf{z}_{i+1}), \sigma_{\theta_i}(\mathbf{z}_{i+1}))$  and  $p_0(\mathbf{z}_L) \sim \mathcal{N}(0, I)$ . Although such a modelling strategy has seen significant improvement in data density estimation, it can be less effective in learning hierarchical representations.

To demonstrate, we train BIVA<sup>1</sup> [21] on MNIST with 6 layers of latent variables and conduct hierarchical sampling by generating latent variables via the reparametrization trick, i.e.,  $\mathbf{z}_i = \mu_{\theta_i}(\mathbf{z}_{i+1}) + \sigma_{\theta_i}(\mathbf{z}_{i+1}) \cdot \epsilon_i$ , where  $\epsilon_i$  is the added Gaussian noise. Specifically, the latent variables are generated by randomly sampling  $\epsilon_i$  for the  $i$ -th layer, while fixing  $\epsilon_{j \neq i}$  for other layers, and the corresponding image synthesis should deliver the variation of representation learned by  $\mathbf{z}_i$ . We show the results in Fig. 1, where the major semantic variation is only captured by the top layer

of the latent variables, suggesting an ineffective hierarchical representation learned (compare with our model shown in Fig. 5).

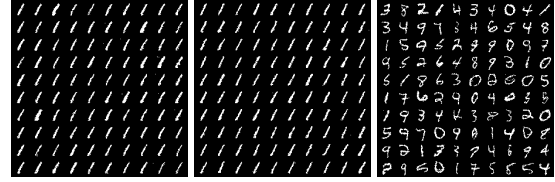


Figure 1. Visualization of hierarchical representations on BIVA. **Left:** sampling  $\epsilon_1, \epsilon_2$  for bottom layers. **Middle:** sampling  $\epsilon_3, \epsilon_4$  for middle layers. **Right:** sampling  $\epsilon_5, \epsilon_6$  for top layers.

### 2.2. Latent space EBM

The energy-based models (EBMs) [7, 6, 24, 8, 38] present a flexible and appealing way for data modelling and are known for being expressive and powerful in capturing data regularities. Albeit their expressivity, learning the EBM on high-dimensional  $\mathbf{x}$ -space can be challenging since the energy function needs to support the entire data space. As such, another line of works [26, 1, 37] seek to build EBM on *low-dimensional*  $\mathbf{z}$ -space with the probability density defined as

$$p_\theta(\mathbf{z}) = \frac{1}{Z(\theta)} \exp[f_\theta(\mathbf{z})] p_0(\mathbf{z}) \quad (1)$$

However, with a single-layer non-hierarchical  $\mathbf{z}$ -space, it is typically *infeasible* to capture the data abstraction of different levels. To see this, we pick up the digit classes ‘0’, ‘1’, and ‘2’ of the MNIST dataset, on which we train LEBM [26] with the latent dimension set to be 2. We show in Fig. 2 that changing the value of each latent unit could lead to a *mixed* variations in digit shape and class (compare to our proposed method in Fig. 6).

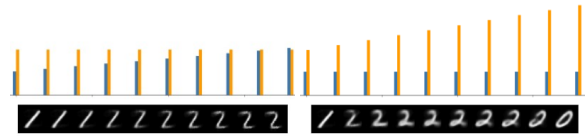


Figure 2. Visualization for latent space EBM [26] by changing each unit of 2-dimensional  $\mathbf{z}$ . **Left:** changing the first unit. **Right:** changing the second unit. **Top:** the value of each unit, where the orange color indicates the first unit and the blue color indicates the second unit.

## 3. Methodology

In this paper, we study a joint latent space EBM prior where latent variables are partitioned into multiple groups but modelled jointly for hierarchical representation learning. To ensure efficient inference, we introduce an inference model to approximate the generator posterior and develop a variational joint training scheme in which each model can be trained efficiently and effectively.

<sup>1</sup><https://github.com/vlievin/biva-pytorch>

### 3.1. Latent variable generative model

A latent variable generative model (i.e., generator model) can be specified using joint distribution:

$$p_{\beta, \alpha}(\mathbf{x}, \mathbf{z}) = p_{\beta}(\mathbf{x}|\mathbf{z})p_{\alpha}(\mathbf{z}) \quad (2)$$

which consists of the generation model  $p_{\beta}(\mathbf{x}|\mathbf{z})$  and the prior model  $p_{\alpha}(\mathbf{z})$ .

**Joint latent space EBM:** The prior model  $p_{\alpha}(\mathbf{z})$  is usually assumed to be single-layer, which can be infeasible for learning hierarchical representations (see Sec. 2.2). In this work, we propose a joint latent space EBM prior, where latent variables are separated into multiple groups, i.e.,  $\mathbf{z} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_L]$ , and our EBM prior model is defined as

$$p_{\alpha}(\mathbf{z}) = \frac{1}{\mathbb{Z}(\alpha)} \exp[f_{\alpha}([\mathbf{z}_1, \dots, \mathbf{z}_L])]p_0([\mathbf{z}_1, \dots, \mathbf{z}_L]) \quad (3)$$

where  $[\cdot]$  refers to the concatenation.  $f_{\alpha}(\cdot)$  is the negative energy function that can be parameterized by a small multi-layer perceptron with the parameters  $\alpha$ ,  $p_0([\cdot])$  is a reference distribution assumed to be a unit Gaussian, and  $\mathbb{Z}(\alpha)$  is the normalizing constant or partition function. Such a prior model can be considered as an energy-based refinement of the non-informative reference distribution, thus the relationship between different groups of latent codes can be well captured via the energy function  $f_{\alpha}([\mathbf{z}_1, \dots, \mathbf{z}_L])$ .

For notation simplicity, we denote  $\mathbf{z} = [\mathbf{z}_1, \dots, \mathbf{z}_L]$  in subsequent discussions.

**Generation Model.** The generation model is defined as

$$p_{\beta}(\mathbf{x}|\mathbf{z}) \sim \mathcal{N}(g_{\beta}(\mathbf{z}), \sigma^2 I_D) \quad (4)$$

which models the observed data using top-down generator network  $g_{\beta}(\mathbf{z})$  with additive Gaussian noise, i.e.,  $\mathbf{x} = g_{\beta}(\mathbf{z}) + \epsilon$ ,  $\epsilon \sim \mathcal{N}(0, \sigma^2 I_D)$ . To facilitate the hierarchical representation learning with multi-layer latent variables, we consider multi-layer hierarchical generator network  $g_{\beta}$  ( $= \{g_1, g_2, \dots, g_L\}$ ) that is designed to explain the observation  $\mathbf{x}$  by integrating data representation from the above layers, i.e.,

$$\begin{aligned} h_L &= g_L(\mathbf{z}_L) \\ h_i &= g_i([\mathbf{z}_i, h_{i+1}]), \quad i = 1, 2, \dots, L-1 \\ \mathbf{x} &\sim \mathcal{N}(h_1, \sigma^2 I_D) \end{aligned} \quad (5)$$

in which  $\mathbf{z}_L$  is at the top layer, and  $g_i$  is a shallow network that decodes latent code  $\mathbf{z}_i$  while integrating features from the upper layer.

**Joint EBM prior vs. independent Gaussian prior:** With such a multi-layer hierarchical generation model, it is also tempting to consider learning with the unit Gaussian prior, i.e.,  $p(\mathbf{z}) \sim \mathcal{N}(0, I_d)$ . Such a prior choice is adopted in

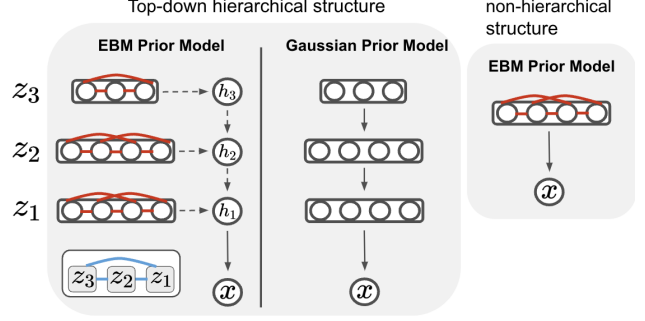


Figure 3. The illustration of the proposed joint EBM prior model (Left). **Red lines** indicates the modelling of *intra-layer* relation, and **blue lines** indicate *inter-layer* relation. Compared to the Gaussian prior model and single-layer LEBM, our prior model is expressive and effective in learning hierarchical representations.

[42, 20] and has seen some success in learning hierarchical features. However, the independent Gaussian prior does not account for the relations between levels of representations and can be less expressive in capturing complex data features. The proposed EBM prior is expressive and can couple the latent variables across different layers within a joint modelling framework.

Our model is illustrated in the left panel of Fig. 3.

### 3.2. Variational learning scheme

Suppose we have observed examples  $(\mathbf{x}_i, i = 1, \dots, n)$  that come from (true) data distribution  $p_{\text{data}}(\mathbf{x})$  and denote  $\theta = (\beta, \alpha)$  that collects all parameters from the generator model. Learning the generator model can be done by maximizing the log-likelihood of all observed examples as

$$\sum_{i=1}^n \log p_{\theta}(\mathbf{x}_i) = \sum_{i=1}^n \log \int p_{\alpha}(\mathbf{z})p_{\beta}(\mathbf{x}_i|\mathbf{z})d\mathbf{z} \quad (6)$$

When  $n$  becomes sufficiently large, maximizing the above log-likelihood is equivalent to minimizing the Kullback-Leibler (KL) divergence between the model distribution and data distribution, i.e.,  $\min_{\theta} D_{\text{KL}}(p_{\text{data}}(\mathbf{x})||p_{\theta}(\mathbf{x}))$ .

However, directly maximizing Eq. 6 can be challenging, as the inference of generator posterior (by chain rule,  $p_{\theta}(\mathbf{z}|\mathbf{x}) = p_{\theta}(\mathbf{x}, \mathbf{z})/p_{\theta}(\mathbf{x})$ ) is typically intractable. To tackle the challenge, we employ an inference model  $q_{\phi}(\mathbf{z}|\mathbf{x})$  with a separate set of parameters  $\phi$  to approximate the generator posterior.

**Inference model:** The inference model maps from data space to latent space and is typically assumed to be Gaussian distributed,

$$q_{\phi}(\mathbf{z}|\mathbf{x}) \sim \mathcal{N}(\mu_{\phi}(\mathbf{x}), V_{\phi}(\mathbf{x})) \quad (7)$$

where  $\mu_{\phi}(\mathbf{x})$  and  $V_{\phi}(\mathbf{x})$  are the d-dimensional mean vector and diagonal covariance matrix. To match with the hierar-

chical generator model and facilitate the hierarchical representation learning, we consider the multi-layer inference model defined as

$$\begin{aligned} r_1 &= h_1(\mathbf{x}) \\ r_i &= h_i(r_{i-1}) \quad i = 2, 3, \dots, L \\ \mathbf{z}_i &\sim \mathcal{N}(\mu_i(r_i), V_i(r_i)) \end{aligned} \quad (8)$$

where  $\mathbf{z}_L$  is the inferred latent code at the top layer and  $\mathbf{z}_1$  is the inferred bottom layer latent code. Each latent code  $\mathbf{z}_i$  can be inferred and sampled through reparametrization trick [17] using  $\mathbf{z}_i = \mu_i(r_i) + V_i(r_i)^{1/2}\epsilon$  where  $\epsilon \sim \mathcal{N}(0, I_{d_i})$ , with  $d_i$  being the dimensionality of  $\mathbf{z}_i$ .

**Variational learning:** We show that the proposed generator model and the inference model can be naturally unified and jointly trained within a variational learning scheme.

Specifically, we can define two joint densities, one for generator model density, i.e.,  $p_\theta(\mathbf{x}, \mathbf{z}) = p_\alpha(\mathbf{z})p_\beta(\mathbf{x}|\mathbf{z})$ , and one for data density, i.e.,  $q_\phi(\mathbf{x}, \mathbf{z}) = p_{\text{data}}(\mathbf{x})q_\phi(\mathbf{z}|\mathbf{x})$ . We propose joint learning through KL minimization, and we denote the objective to be  $L(\theta, \phi)$ , i.e.,

$$\min_{\theta} \min_{\phi} L(\theta, \phi) = \min_{\theta} \min_{\phi} D_{\text{KL}}(q_\phi(\mathbf{x}, \mathbf{z}) || p_\theta(\mathbf{x}, \mathbf{z})) \quad (9)$$

where the generator model density is learned to match the true joint data density and capture extracted hierarchical features.

**Learning  $p_\alpha(\mathbf{z})$ :** With this joint KL minimization, the prior model can be learned with the gradient  $-\nabla_\alpha L(\theta, \phi)$

$$\begin{aligned} \mathbb{E}_{q_\phi(\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_L | \mathbf{x})} [\nabla_\alpha f_\alpha([\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_L])] - \\ \mathbb{E}_{p_\alpha(\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_L)} [\nabla_\alpha f_\alpha([\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_L])] \end{aligned} \quad (10)$$

**Learning  $p_\beta(\mathbf{x}|\mathbf{z})$ :** The generation model can be learned with the gradient  $-\nabla_\beta L(\theta, \phi)$  computed as

$$\mathbb{E}_{q_\phi(\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_L | \mathbf{x})} [\nabla_\beta \log p_\beta(\mathbf{x} | \mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_L)] \quad (11)$$

**Learning  $q_\phi(\mathbf{z}|\mathbf{x})$ :** The inference model can be learned by computing the gradient  $-\nabla_\phi L(\theta, \phi)$  as

$$\begin{aligned} \nabla_\phi \mathbb{E}_{q_\phi(\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_L | \mathbf{x})} [\log p_\beta(\mathbf{x} | \mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_L)] - \\ \nabla_\phi D_{\text{KL}}(q_\phi(\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_L | \mathbf{x}) || p_\alpha(\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_L)) \end{aligned} \quad (12)$$

We provide further theoretical understanding in Sec. 3.4.

### 3.3. Prior sampling

Training the proposed joint EBM prior model requires sampling  $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_L$  from the proposed EBM prior (see Eq. 10). To do so, we can use Langevin dynamic [22].

For arbitrary target distribution  $p(\mathbf{z})$ , the Langevin dynamic iterates:

$$\mathbf{z}_{t+1} = \mathbf{z}_t + \frac{s}{2} \nabla_{\mathbf{z}} \log p(\mathbf{z}_t) + s\epsilon_t \quad (13)$$

where  $t$  indexes the time step of Langevin dynamics.  $s$  is the step size and  $\epsilon_t \sim \mathcal{N}(0, I_d)$ . We run the noise-initialized Langevin dynamics for  $K$  steps. Noted that as  $s \rightarrow 0$ , and  $K \rightarrow \infty$ , the distribution of  $\mathbf{z}_t$  will converge to the target  $p(\mathbf{z})$  regardless of the initial distribution of  $\mathbf{z}_0$  [22].

Specifically, for prior sampling, we use the prior distribution  $p_\alpha(\mathbf{z})$  as our target distribution, and the gradient  $\nabla_{\mathbf{z}} \log p_\alpha(\mathbf{z})$  is computed as:

$$\nabla_{\mathbf{z}} f_\alpha([\mathbf{z}_1, \dots, \mathbf{z}_L]) + \nabla_{\mathbf{z}} \log p_0([\mathbf{z}_1, \dots, \mathbf{z}_L]) \quad (14)$$

The Langevin dynamics for prior sampling can be efficient due to the low-dimensional latent space and the lightweight network structure for the energy function.

The overall procedure is summarized in Algorithm 1.

---

**Algorithm 1** Learning latent EBM prior for hierarchical generator model

---

**Input:** Training iterations  $T$ , observed training examples  $\{x_i\}_{i=1}^n$ , batch size  $m$ , network parameters  $\alpha, \beta, \phi$ , learning rate  $\eta_\alpha, \eta_\beta, \eta_\phi$ , Langevin steps  $K$ , Langevin step size  $s$ ,

Let  $t = 0$  and  $\theta = (\alpha, \beta)$ ;

**repeat**

**Prior sampling** for  $\{z_i^-\}_{i=1}^m$  using Eq. 13 and Eq. 14 with  $K$  and  $s$ .

**Inference sampling** for  $\{z_i^+\}_{i=1}^m$  using Eq. 8 with  $\{x_i\}_{i=1}^m$  and reparametrization.

**Learn joint EBM prior:** Given  $\{z_i^-, z_i^+\}_{i=1}^m$ , update  $\alpha = \alpha - \eta_\alpha \nabla_\alpha L(\theta, \phi)$  using Eq. 10 with  $\eta_\alpha$

**Learn generation model:** Given  $\{x_i, z_i^+\}_{i=1}^m$  update  $\beta = \beta - \eta_\beta \nabla_\beta L(\theta, \phi)$  using Eq. 11 with  $\eta_\beta$

**Learn inference model:** Given  $\{x_i, z_i^+\}_{i=1}^m$  update  $\phi = \phi - \eta_\phi \nabla_\phi L(\theta, \phi)$  using Eq. 12 with  $\eta_\phi$

    Let  $t = t + 1$ ;

**until**  $t = T$

---

### 3.4. Theoretical understanding

**Divergence perturbation and ELBO.** The KL joint minimization (Eq. 9) can be viewed as a surrogate of the MLE objective with the KL perturbation term,

$$\begin{aligned} D_{\text{KL}}(q_\phi(\mathbf{x}, \mathbf{z}) || p_\theta(\mathbf{x}, \mathbf{z})) \\ = D_{\text{KL}}(p_{\text{data}}(\mathbf{x}) || p_\theta(\mathbf{x})) + D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z}|\mathbf{x})) \end{aligned}$$

where  $D_{\text{KL}}(p_{\text{data}}(\mathbf{x}) || p_\theta(\mathbf{x}))$  is the MLE loss function, and the perturbation term  $D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z}|\mathbf{x}))$  measures the KL-divergence between inference distribution and generator posterior. The inference  $q_\phi(\mathbf{z}|\mathbf{x})$  is learned to match the posterior distribution of the generator without expensive posterior sampling. In fact, such KL minimization in the joint space is equivalent to evidence lower bound



(ELBO). To see this, noted that  $D_{\text{KL}}(p_{\text{data}}(\mathbf{x})||p_{\theta}(\mathbf{x})) = -H(p_{\text{data}}(\mathbf{x})) - \mathbb{E}_{p_{\text{data}}}[\log p_{\theta}(\mathbf{x})]$  where  $H(p_{\text{data}}(\mathbf{x}))$  is the entropy of the empirical data distribution and can be treated as constant  $C \equiv -H(p_{\text{data}}(\mathbf{x}))$  w.r.t model parameters  $\theta$ . Then  $L(\theta, \phi)$  is computed as

$$\begin{aligned} & -\mathbb{E}_{p_{\text{data}}}[\log p_{\theta}(\mathbf{x})] + D_{\text{KL}}(q_{\phi}(\mathbf{z}|\mathbf{x})||p_{\theta}(\mathbf{z}|\mathbf{x})) + C \\ &= \mathbb{E}_{p_{\text{data}}} \left[ \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} \left( \log \frac{q_{\phi}(\mathbf{z}|\mathbf{x})}{p_{\theta}(\mathbf{z}|\mathbf{x})} \right) - \log p_{\theta}(\mathbf{x}) \right] + C \\ &= \mathbb{E}_{p_{\text{data}}} [-\text{ELBO}(\mathbf{x}; \theta, \phi)] + C \end{aligned}$$

where

$$\begin{aligned} \text{ELBO}(\mathbf{x}; \theta, \phi) &= \log p_{\theta}(\mathbf{x}) - \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} \left[ \log \frac{q_{\phi}(\mathbf{z}|\mathbf{x})}{p_{\theta}(\mathbf{z}|\mathbf{x})} \right] \\ &= \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} \left[ \log \frac{p_{\theta}(\mathbf{x}, \mathbf{z})}{q_{\phi}(\mathbf{z}|\mathbf{x})} \right] \end{aligned}$$

and is referred to as evidence lower bound (ELBO) in the literature [17]. Noted that the prior model (i.e.,  $p_{\theta}(\mathbf{x}, \mathbf{z}) = p_{\beta}(\mathbf{x}|\mathbf{z})p_{\alpha}(\mathbf{z})$ ) is now parameterized to be our joint EBM prior. Minimizing Eq. 9 is equivalent to maximizing the evidence lower bound of the log-likelihood.

**Short run sampling and divergence perturbation.** The learning of the energy-based model in Eq. 10 requires MCMC sampling from  $p_{\alpha}(\mathbf{z})$ . We adopt the short-run Langevin dynamics (Eq. 13) with fixed  $K$  steps to sample from prior  $p_{\alpha}(\mathbf{z})$  for efficient computation. Such prior sampling amounts to approximates the objective  $L(\theta, \phi)$  in Eq. 9 with yet another KL perturbation term, i.e.,

$$\tilde{L}(\theta, \phi) = L(\theta, \phi) - D_{\text{KL}}(\tilde{p}_{\alpha(t)}(\mathbf{z})||p_{\alpha}(\mathbf{z}))$$

where  $\tilde{p}_{\alpha(t)}(\mathbf{z})$  refers to the distribution of latent codes after  $K$  steps of Langevin dynamics (Eq. 13) in the  $t$ -th iteration of the learning. The corresponding learning gradient for the prior model is thus

$$-\nabla_{\alpha} \tilde{L}(\theta, \phi) = \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} [\nabla_{\alpha} f_{\alpha}(\mathbf{z})] - \mathbb{E}_{\tilde{p}_{\alpha(t)}(\mathbf{z})} [\nabla_{\alpha} f_{\alpha}(\mathbf{z})]$$

When  $K$  is sufficiently large, the perturbation term  $D_{\text{KL}}(\tilde{p}_{\alpha(t)}(\mathbf{z})||p_{\alpha}(\mathbf{z})) \rightarrow 0$ . The update rule for the energy-based prior model (also Eq. 10) can be interpreted as self-adversarial learning. The prior model  $p_{\alpha}(\mathbf{z})$  serves as both generator and discriminator if we compare it to GAN (generative adversarial networks) [11]. In contrast to GAN, our learning follows MLE with divergence perturbations, which in general does not suffer from issues such as mode collapsing and instability, as it does not involve competition between two separate models. The energy-based model can thus be seen as its own adversary or its own critic.

## 4. Related Work

**Generator model.** The generator model has a top-down network that maps the low-dimensional latent code to the

observed data space. VAEs [17, 27, 33, 32] proposes variational learning by introducing an approximation of the true intractable posterior, which allows a tractable bound on log-likelihood to be maximized. Another line of work [14, 25] trains the generator model without inference model by using Langevin dynamics for generator posterior sampling. We follow the former approach and use the inference model for efficient training.

**Generator model with informative prior.** The majority of the existing generator models assume that the latent code follows a simple and known prior distribution, such as isotropic Gaussian distribution. Such assumption may render an ineffective generator as observed in [34, 5]. Our work utilizes the informative and expressive energy-based prior in the joint space of latent variables and is related to the line of previous research on introducing flexible prior distributions. For example, [34] parameterizes the prior based on the posterior inference model. [5, 9] adopt a two-stage approach which first trains a latent variable model with simple prior and then trains a separate prior model (e.g., VAE or Gaussian mixture model) to match the aggregated posterior distribution.

**Generator model with hierarchical structures.** The generator models that consist of multi-layers of latent variables are recently attracting attention for hierarchical learning. [29, 2, 21, 3, 36] stack generative models on top of each other and assume conditional Gaussian distributions at different layers. However, as shown in Sec. 2.1, the bottom layer could act as a single generative model and absorb most features, leading to ineffective representation learning. [42, 20] propose to learn hierarchical features with unit Gaussian prior by adapting the generator into multiple layers. However, the latent variables at different layers are distributed independently *a priori* and are only loosely connected. Instead, we propose a joint EBM prior that could tightly couple and jointly model latent variables at different layers and show superior results through joint training.

### Hierarchical generator model with informative prior.

Recent advances [4, 1] have started exploring learning informative prior for multi-layer generator models. JEBM [4] and NCP-VAE [1] build latent space EBM on hierarchical generator models and showcase the capability of improving the generation quality. However, these models consider the *conditional hierarchical models*, which can be limited in learning effective hierarchical representations (see Sec. 2.1). This work studies hierarchical representation learning by introducing a novel framework where latent space EBM can be jointly trained with the *architectural hierarchical models* to capture complex representations of different levels.



Figure 4. Hierarchical sampling on SVHN. **Left:** The latent code at bottom layer ( $z_1$ ) represents the background light and shading. **Center-left:** the latent code at second bottom layer ( $z_2$ ) represents the color schemes. **Center-right:** the latent code at second top layer ( $z_3$ ) encodes the shape variations of the same digit. **Right:** the latent code at top layer ( $z_4$ ) captures the digit identity and the general structure.



Figure 5. Hierarchical sampling on MNIST. **Left:** The latent code at bottom layer ( $z_1$ ) indicates the stroke width. **Center:** the latent code at second layer ( $z_2$ ) encodes geometric changes among similar digits. **Right:** the latent code at top layer ( $z_3$ ) learns the digit identity and general structure.

## 5. Experiments

To demonstrate the effectiveness of our method, we conduct the following experiments: 1) hierarchical representation learning, 2) analysis of latent space, 3) image modelling, and 4) robustness. For a better understanding of the proposed method, we perform the disentanglement learning experiment, various ablation studies, and an analysis of parameter complexity.

### 5.1. Hierarchical Representation Learning

We first demonstrate the capability of our model in hierarchical representation learning by experiments of hierarchical sampling and latent classifier.

**Hierarchical sampling.** With partitioned latent variables and the hierarchical structure of generator, the latent variables at higher layers should capture semantic features, while the lower layers should capture low-level features. We train our model on MNIST using the same setting as in Sec. 2.2 with only ‘0’, ‘1’, and ‘2’ digit classes available and the latent dimension set to 2. In Fig. 6, by changing the  $z_1$  (at the bottom layer), our model can successfully deliver the variations in shape (low-level features), and changing the  $z_2$  (at the top layer) leads to the variations in digit class (high-level features). This indicates a successfully learned hierarchical representation.

Further, we train our model on MNIST with all digit

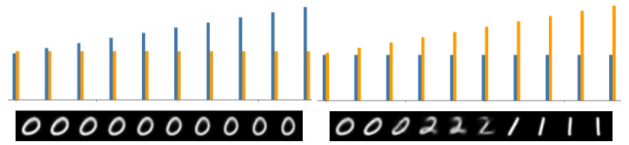


Figure 6. Visualization for our model by changing each unit of 2-dimensional  $z$ . **Left:** changing the first unit. **Right:** changing the second unit. **Top:** the value of each unit, where the orange color indicates the first unit and the blue color indicates the second unit.

classes and the more challenging SVHN dataset, and we generate the images by sampling (using Eq. 13) the latent code of one layer while keeping other layers fixed. The results shown in Fig. 4 and Fig. 5 also demonstrate the hierarchical representation learned by our model.

**Latent Classifier.** We next show that our model can achieve better performance in hierarchical representation learning. Note that the inference model  $q_\phi(z|x)$  should be learned to approximate the true posterior  $p_\theta(z|x)$  of the generator, which integrates both the hierarchical inductive bias and the energy-based refinement. Therefore, to measure the learned hierarchical features, we learn classifiers on the inferred latent codes at different layers to predict the label of the data and measure the testing classification accuracy. The latent codes that carry rich high-level semantic features should achieve higher classification accuracy.

Table 1. Testing accuracy of inferred latent codes at each layer. We denote ( $\downarrow$ ) for the layer that has **unexpectedly lower** accuracy.

		$L = 1$	$L = 2$	$L = 3$	$L = 4$
MNIST	Ours	32.60%	58.32%	67.64%	-
	VLAE	35.69%	57.86%	52.08% ( $\downarrow$ )	-
Fashion-MNIST	Ours	39.27%	44.81%	84.27%	-
	VLAE	52.43%	35.65% ( $\downarrow$ )	83.24%	-
SVHN	Ours	21.00%	25.29%	30.58%	86.63%
	VLAE	22.59%	26.97%	56.70%	52.14% ( $\downarrow$ )

In practice, we first train our model on standard benchmarks, such as MNIST, Fashion-MNIST, and SVHN. Then, we train classifiers on the inferred latent codes at each layer. We consider VLAE [42] as our baseline model, which

learns a structural latent space with standard Gaussian prior. We show comparison results in Tab. 1, where our learned inference model  $q_\phi(\mathbf{z}|\mathbf{x})$  extracts varying levels of semantic features for different layers of latent codes. The top layer carries the most significant features, and the bottom layer carries the least. The baseline model that assumes Gaussian prior, however, extracted mixed semantic features across different layers. For fair comparisons, we use the same generation and inference structure as the baseline model<sup>2</sup>. The structure of the classifier contains two linear layers with a hidden dimension set to 256 and uses ReLU activation function. The classifier has softmax as its output layer. We do not employ drop-out and normalization strategies.

## 5.2. Image Modelling

In this section, we evaluate our model in image modelling by measuring the quality of generated and reconstructed images. We consider the baseline models that assume Gaussian prior, such as Alternating Back-propagation (ABP) [14], Ladder-VAE (LVAE) [30], BIVA [21], Short-run MCMC (SRI) [25], and VLAE [42], as well as generator models with informative prior, such as RAE [9], Two-stages VAE (2s-VAE) [5], NCP-VAE [1], Multi-NCP [37] and LEBM [26]. To make fair comparisons, we follow the protocol in [26].

Table 2. Testing reconstruction by MSE, and generation evaluation by FID on SVHN and CelebA-64.

Model	SVHN		CelebA-64	
	MSE ( $\downarrow$ )	FID ( $\downarrow$ )	MSE ( $\downarrow$ )	FID ( $\downarrow$ )
ABP	-	49.71	-	51.50
LVAE	0.014	39.26	0.028	53.40
BIVA	0.010	31.65	0.010	33.58
SRI	0.011	35.23	0.011	36.84
VLAE	0.016	43.95	0.010	44.05
2s-VAE	0.019	42.81	0.021	44.40
RAE	0.014	40.02	0.018	40.95
NCP-VAE	0.020	33.23	0.021	42.07
Multi-NCP	<b>0.004</b>	26.19	0.009	35.38
LEBM	0.008	29.44	0.013	37.87
Ours	0.008	<b>24.16</b>	<b>0.004</b>	<b>32.15</b>

**Synthesis:** We use Fréchet Inception Distance (FID) [16] to quantitatively evaluate the sample quality. In Tab. 2, it can be seen that our model obtains superior generation quality compared to the baseline models.

**Reconstructions:** To evaluate the accuracy of our inference model, we compute mean square error (MSE) [26, 24] for testing reconstruction. As shown in Tab. 2, the proposed model leads to more accurate and faithful reconstructions.

<sup>2</sup><https://github.com/ermongroup/Variational-Ladder-Autoencoder>

## 5.3. Analysis of Latent Space

**Visualization of toy data.** To illustrate the expressivity of the proposed EBM prior, we train our model on MNIST (using the same setting as shown in Fig. 2 and Fig. 6) and set the latent dimension for both  $\mathbf{z}_1$  and  $\mathbf{z}_2$  to 2 for better visualization. We visualized the Langevin transition of the prior sampling in Fig. 7, which shows that the latent codes at the top layer ( $\mathbf{z}_2$ ) can effectively capture the multi-modal posterior, where using a Gaussian prior might be infeasible.

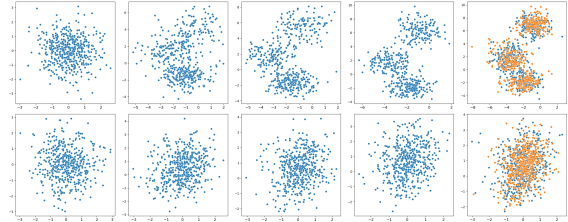


Figure 7. Visualization of the latent codes sampled from our EBM prior (Top row:  $\mathbf{z}_2$ ). **Blue, Orange** color indicate prior and posterior, respectively.

**Langevin trajectory.** We explore the energy landscape of the learned prior  $p_\alpha(\mathbf{z})$  via transition of Langevin dynamics initialized from  $p_0(\mathbf{z})$  towards  $p_\alpha(\mathbf{z})$  on CelebA-64. If the EBM is well-learned, the energy prior should render local modes of the energy function, and traversing these local modes should present diverse, realistic image synthesis and steady-state energy scores. Existing EBMs often suffer from oversaturated synthesis via the challenging *long-run* Langevin dynamic (see oversaturated example in Fig.3 in [23]). We thereby test our model with 2500 Langevin steps, which is much longer than the 40 steps used in training. Fig. 8 shows the image synthesis and energy profile, where our EBM prior model does not exhibit the oversaturated phenomenon and successfully delivers diverse and realistic synthesis with steady-state energy.

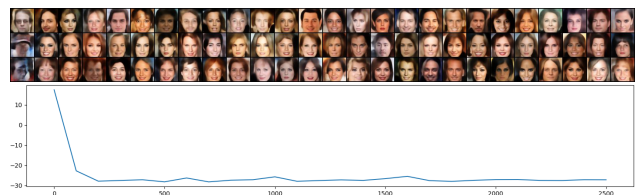


Figure 8. Transition of Markov chains initialized from  $p_0(\mathbf{z})$  towards  $p_\alpha(\mathbf{z})$  for 2500 steps. Top: Trajectory in the CelebA-64 data space for every 100 steps. Bottom: Energy profile over time.

## 5.4. Robustness

In this section, we examine the robustness of our model in adversarial attack and outlier detection.

**Adversarial robustness.** For adversarial robustness, we consider the latent attack [10], which is shown to be a



strong attack for the generative model with latent variables, and it allows us to attack the latent codes at different layers. Specifically, let  $\mathbf{x}_t$  be the target and  $\mathbf{x}_o$  be the original image. The attack aims to create the distorted image  $\tilde{\mathbf{x}} = \mathbf{x}_o + \epsilon$  such that its learned inference distribution  $q_\phi(\mathbf{z}|\tilde{\mathbf{x}})$  is close to the inference  $q_\phi(\mathbf{z}|\mathbf{x}_t)$  on target images under KL-divergence (Eq. 5 in [10]).

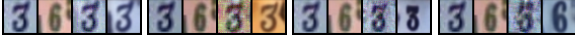


Figure 9. Adversarial latent attack on different layers. The leftmost figure indicates the bottom layer, and the rightmost figure indicates the top layer. **1st col:** original image. **2nd col:** target image. **3rd col:** adversarial example. **4th col:** reconstruction of adversarial example.

We illustrate the adversarial attack on different layers in Fig. 9. The learned inference model  $q_\phi(\mathbf{z}|\mathbf{x})$  carries different levels of semantics for different layers. The adversarial example created by attacking latent codes at the bottom layer only perturbs the data along low-level features direction from the target image; hence, the model can be robust and have faithful reconstruction from adversarial examples. The adversarial example coming from the top-layer attack can distort the original data along a high-level semantic direction from the target image, thus, the reconstruction from the adversarial example can be substantially different from the original data, indicating a successful attack.

**Anomaly detection.** We also evaluate the model on anomaly detection. If the model is well-learned, the inference model  $q_\phi(\mathbf{z}|\mathbf{x})$  should form an informative latent space that separates anomalies from the normal data. Following the protocols in [18, 26], we take latent samples from the learned inference model and use the unnormalized log-posterior  $\log p_\theta(\mathbf{x}, \mathbf{z})$  as our decision function to detect anomaly on MNIST. Only one digit class is held-out as an anomaly in training, while both normal and anomalous data are used for testing. We compare baseline models such as BiGAN- $\sigma$  [41], MEG [18], LEBM [26], HVAE and VAE using area under the precision-recall curve (AUPRC). The results of baseline models are provided by [26], and we use  $L^{>0}$  [15] as the decision function for HVAE model. Tab. 3 shows the result.

Table 3. AUPRC scores for unsupervised anomaly detection on MNIST. Following the protocol of [26], we average the results over last 10 epochs to account for variance.

hold-out digit	1	4	5	7	9
VAE	0.063	0.337	0.325	0.148	0.104
HVAE	0.494 $\pm$ 0.004	0.920 $\pm$ 0.004	0.913 $\pm$ 0.003	0.680 $\pm$ 0.006	0.791 $\pm$ 0.008
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- $\sigma$	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
Latent EBM	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
Ours	<b>0.722 <math>\pm</math> 0.010</b>	<b>0.949 <math>\pm</math> 0.002</b>	<b>0.980 <math>\pm</math> 0.001</b>	<b>0.941 <math>\pm</math> 0.003</b>	<b>0.935 <math>\pm</math> 0.003</b>

## 5.5. Ablation Studies

The proposed expressive prior model enables complex data representations to be effectively captured, which in turn

improves the expressivity of the whole model in generating high-quality synthesis. To better understand the influence of the proposed method, we conduct ablation studies, including 1) MCMC steps, 2) complexity of EBM, and 3) other image datasets.

**MCMC steps:** We examine the influence of steps of short-run MCMC for prior sampling. Tab. 4 shows that increasing steps could result in a better quality of generation. Increasing the step number from 15 to 60 could result in a significant improvement in synthesis quality, while exhibiting only minor influence when increased beyond 60.

Table 4. Varying  $k$  (steps number) of our model on CelebA-64.

	$k=15$	$k=30$	$k=60$	$k=150$	$k=300$
FID	56.42	39.89	<b>32.15</b>	31.20	30.78

**Complexity of EBM:** We further examine the influence of model parameters of EBM. The energy function is parameterized by a 2-layer perceptron. We fix the number of layers and increase the hidden units of EBM. As shown in Tab. 5, increasing the hidden units of EBM results in better performance.

Table 5. Increasing hidden units (denoted as **nef**) of EBM.

nef	nef= 0	nef= 10	nef= 20	nef= 50	nef= 100
FID	43.95	41.72	33.10	30.42	<b>24.16</b>

**Other image datasets:** We test the scalability of our model on high-resolution image dataset, such as CelebA-128. We show the image synthesis in Fig. 10.

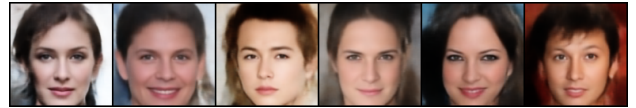


Figure 10. Generated image synthesis on CelebA-128

We also train our model on CIFAR-10 (32x32) and report the FID score for generation quality in Tab. 6, in which our model performs well compared to baseline models.

Table 6. FID on CIFAR-10

Ours	2s-VAE	RAE	NCP-VAE	Multi-NCP	LEBM
63.42	72.90	74.16	78.06	65.01	70.15

## 6. Conclusion

In this paper, we propose to build a joint latent space EBM with hierarchical structures for better hierarchical representation learning. Our joint EBM prior is expressive in modelling the inter-layer relation among multiple layers of latent variables, which in turn enables different levels of data representations to be effectively captured. We develop an efficient variational learning scheme and conduct various experiments to demonstrate the capability of our model in hierarchical representation learning and image modelling. For future work, we shall explore learning the hierarchical representations from other domains, such as videos and texts.



## References

- [1] Jyoti Aneja, Alex Schwing, Jan Kautz, and Arash Vahdat. A contrastive learning approach for training variational autoencoder priors. *Advances in neural information processing systems*, 34:480–493, 2021. 2, 5, 7
- [2] Philip Bachman. An architecture for deep, hierarchical generative models. *Advances in Neural Information Processing Systems*, 29, 2016. 5
- [3] Rewon Child. Very deep vaes generalize autoregressive models and can outperform them on images. *arXiv preprint arXiv:2011.10650*, 2020. 1, 5
- [4] Jiali Cui, Ying Nian Wu, and Tian Han. Learning joint latent space ebm prior model for multi-layer generator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3603–3612, 2023. 1, 5
- [5] Bin Dai and David Wipf. Diagnosing and enhancing vae models. *arXiv preprint arXiv:1903.05789*, 2019. 5, 7
- [6] Yilun Du, Shuang Li, Joshua Tenenbaum, and Igor Mordatch. Improved contrastive divergence training of energy based models. *arXiv preprint arXiv:2012.01316*, 2020. 2
- [7] Yilun Du and Igor Mordatch. Implicit generation and generalization in energy-based models. *arXiv preprint arXiv:1903.08689*, 2019. 1, 2
- [8] Ruiqi Gao, Yang Song, Ben Poole, Ying Nian Wu, and Diederik P Kingma. Learning energy-based models by diffusion recovery likelihood. *arXiv preprint arXiv:2012.08125*, 2020. 2
- [9] Partha Ghosh, Mehdi SM Sajjadi, Antonio Vergari, Michael Black, and Bernhard Schölkopf. From variational to deterministic autoencoders. *arXiv preprint arXiv:1903.12436*, 2019. 5, 7
- [10] George Gondim-Ribeiro, Pedro Tabacof, and Eduardo Valle. Adversarial attacks on variational autoencoders. *arXiv preprint arXiv:1806.04646*, 2018. 7, 8
- [11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 5
- [12] Ishaan Gulrajani, Kundan Kumar, Faruk Ahmed, Adrien Ali Taiga, Francesco Visin, David Vazquez, and Aaron Courville. Pixelvae: A latent variable model for natural images. *arXiv preprint arXiv:1611.05013*, 2016. 1
- [13] Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. Long text generation via adversarial training with leaked information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, number 1, 2018. 1
- [14] Tian Han, Yang Lu, Song-Chun Zhu, and Ying Nian Wu. Alternating back-propagation for generator network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017. 5, 7
- [15] Jakob D Drachmann Havtorn, Jes Frellsen, Søren Hauberg, and Lars Maaløe. Hierarchical vaes know what they don’t know. In *International Conference on Machine Learning*, pages 4117–4128. PMLR, 2021. 8
- [16] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 7
- [17] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 4, 5
- [18] Rithesh Kumar, Sherjil Ozair, Anirudh Goyal, Aaron Courville, and Yoshua Bengio. Maximum entropy generators for energy-based models. *arXiv preprint arXiv:1901.08508*, 2019. 8
- [19] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006. 1
- [20] Zhiyuan Li, Jaideep Vitthal Murkute, Prashanna Kumar Gyawali, and Linwei Wang. Progressive learning and disentanglement of hierarchical representations. *arXiv preprint arXiv:2002.10549*, 2020. 1, 3, 5
- [21] Lars Maaløe, Marco Fraccaro, Valentin Liévin, and Ole Winther. Biva: A very deep hierarchy of latent variables for generative modeling. *Advances in neural information processing systems*, 32, 2019. 1, 2, 5, 7
- [22] Radford M Neal et al. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2, 2011. 4
- [23] 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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5272–5280, 2020. 7
- [24] 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*, 32, 2019. 1, 2, 7
- [25] Erik Nijkamp, Bo Pang, Tian Han, Linqi Zhou, Song-Chun Zhu, and Ying Nian Wu. Learning multi-layer latent variable model via variational optimization of short run mcmc for approximate inference. In *European Conference on Computer Vision*, pages 361–378. Springer, 2020. 1, 5, 7
- [26] Bo Pang, Tian Han, Erik Nijkamp, Song-Chun Zhu, and Ying Nian Wu. Learning latent space energy-based prior model. *Advances in Neural Information Processing Systems*, 33:21994–22008, 2020. 1, 2, 7, 8
- [27] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014. 5
- [28] Masaki Saito, Eiichi Matsumoto, and Shunta Saito. Temporal generative adversarial nets with singular value clipping. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 1
- [29] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. 5
- [30] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational

- autoencoders. *Advances in neural information processing systems*, 29, 2016. [1](#), [2](#), [7](#)
- [31] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. [1](#)
  - [32] Yuhta Takida, Takashi Shibuya, WeiHsiang Liao, Chieh-Hsin Lai, Junki Ohmura, Toshimitsu Uesaka, Naoki Murata, Shusuke Takahashi, Toshiyuki Kumakura, and Yuki Mitsufuji. Sq-vae: Variational bayes on discrete representation with self-annealed stochastic quantization. *arXiv preprint arXiv:2205.07547*, 2022. [5](#)
  - [33] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558*, 2017. [5](#)
  - [34] Jakub Tomczak and Max Welling. Vae with a vampprior. In *International Conference on Artificial Intelligence and Statistics*, pages 1214–1223. PMLR, 2018. [5](#)
  - [35] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. [1](#)
  - [36] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *Advances in Neural Information Processing Systems*, 33:19667–19679, 2020. [1](#), [2](#), [5](#)
  - [37] Zhisheng Xiao and Tian Han. Adaptive multi-stage density ratio estimation for learning latent space energy-based model. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. [2](#), [7](#)
  - [38] Zhisheng Xiao, Karsten Kreis, Jan Kautz, and Arash Vahdat. Vaebm: A symbiosis between variational autoencoders and energy-based models. *arXiv preprint arXiv:2010.00654*, 2020. [2](#)
  - [39] Jianwen Xie, Yang Lu, Song-Chun Zhu, and Yingnian Wu. A theory of generative convnet. In *International Conference on Machine Learning*, pages 2635–2644. PMLR, 2016. [1](#)
  - [40] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017. [1](#)
  - [41] Houssam Zenati, Chuan Sheng Foo, Bruno Lecouat, Gaurav Manek, and Vijay Ramaseshan Chandrasekhar. Efficient gan-based anomaly detection. *arXiv preprint arXiv:1802.06222*, 2018. [8](#)
  - [42] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Learning hierarchical features from generative models. *arXiv preprint arXiv:1702.08396*, 2017. [1](#), [3](#), [5](#), [6](#), [7](#)