Learning Joint Latent Space EBM Prior Model for Multi-layer Generator

Jiali Cui\textsuperscript{1}, Ying Nian Wu\textsuperscript{2}, Tian Han\textsuperscript{1}
\textsuperscript{1}Department of Computer Science, Stevens Institute of Technology
\textsuperscript{2}Department of Statistics, University of California, Los Angeles
\{jcul7,than6\}@stevens.edu, ywu@stat.ucla.edu

Abstract

This paper studies the fundamental problem of learning multi-layer generator models. The multi-layer generator model builds multiple layers of latent variables as a prior model on top of the generator, which benefits learning complex data distribution and hierarchical representations. However, such a prior model usually focuses on modeling inter-layer relations between latent variables by assuming non-informative (conditional) Gaussian distributions, which can be limited in model expressivity. To tackle this issue and learn more expressive prior models, we propose an energy-based model (EBM) on the joint latent space over all layers of latent variables with the multi-layer generator as its backbone. Such joint latent space EBM prior model captures the intra-layer contextual relations at each layer through layer-wise energy terms, and latent variables across different layers are jointly corrected. We develop a joint training scheme via maximum likelihood estimation (MLE), which involves Markov Chain Monte Carlo (MCMC) sampling for both prior and posterior distributions of the latent variables from different layers. To ensure efficient inference and learning, we further propose a variational training scheme where an inference model is used to amortize the costly posterior MCMC sampling. Our experiments demonstrate that the learned model can be expressive in generating high-quality images and capturing hierarchical features for better outlier detection.

1. Introduction

Deep generative models (a.k.a. generator models) have made promising progress in learning complex data distributions and achieved great successes in image and video synthesis [21, 34, 37, 39] as well as representation learning [5, 48]. Such models usually consist of low-dimensional latent variables together with a top-down generation model that maps such latent factors to the observed data. The latent factors can serve as an abstract data representation, but it is often modelled via a single latent vector with non-informative prior distribution which leads to limited model expressivity and fails to capture different levels of abstractions. Learning an informative prior model for hierarchical representations is needed, yet research in this direction is still under-developed.

A principled way to learn such a prior model is by learning the generator models with multiple layers of latent variables. However, the learning of multi-layer generator model can be challenging as the inter-layer structural relation (i.e., latent variables across different layers) and the intra-layer contextual relation (i.e., latent units within the same layer) have to be effectively modelled and efficiently learned. Various methods have been proposed [5, 28, 32, 35, 40], but they only focused on inter-layer modeling by assuming the conditional Gaussian distribution across different layers while ignoring the intra-layer contextual modeling as the latent units are conditional independent within each layer.

The energy-based models (EBMs), on the other hand, are shown to be expressive and proved to be powerful in capturing contextual and non-structural data regularities. Notably, [33] considers the EBM in the latent space for the non-hierarchical generator model, where the energy function is considered as a correction of the non-informative Gaussian prior. The low dimensionality of the latent space makes EBM effective in capturing regularities in the data. However, a single latent vector in [33] is infeasible for capturing the patterns at multiple layers of abstractions, which limits its model capacity.

In this paper, we propose to combine the strengths of the latent space EBM and the generator with multiple layers of latent variables for better hierarchical representations and a more expressive prior model. Specifically, we introduce layer-wise energy terms to exponentially tilt the non-informative Gaussian conditional at each layer, and latent variables across different layers are modelled jointly through EBM with the multi-layer generator model as its backbone. Such a joint EBM prior model seamlessly integrates the intra-layer contextual modeling via layer-wise energy terms and inter-layer structural modeling with multi-layer latent variables.
The joint EBM prior model can be learned by maximum likelihood estimation (MLE). Each learning iteration involves Markov chain Monte Carlo (MCMC) sampling of latent variables in each layer from both the prior and posterior distributions. The prior sampling can be efficiently done due to the low dimensionality of the latent variables and, more importantly, the lightweight networks for energy functions, while the posterior sampling can be less efficient. Therefore, we further develop the variational training scheme where an additional inference model is used for posterior approximation and is jointly trained with the joint EBM prior model.

Contributions: 1) We propose a joint latent space EBM prior model for the generator model with multiple layers of latent variables; 2) We develop the maximum likelihood learning algorithm that learns the joint EBM prior model based on MCMC prior and posterior sampling across different layers. We further propose the variational joint training scheme for efficient learning and inference; 3) We provide strong empirical results through extensive experiments.

2. Background

In this section, we present the background of multi-layer latent variable model and latent space EBM prior model, which shall serve as the foundation of the proposed model.

2.1. Multi-layer latent variable model

Let \( x \) be the high-dimensional observed example, and \( z \) be the low-dimensional latent variables. The latent variable generative model, or generator model, factorizes a joint distribution of \((x, z)\) as

\[
p_\beta(x, z) = p_{\beta_0}(x|z)p_{\beta_{>0}}(z)
\]

(1)

where \( p_{\beta_0}(x|z) \) is the generation model with parameter \( \beta_0 \) that maps from latent space to data space, and \( p_{\beta_{>0}}(z) \) is the prior distribution over latent variables with parameter \( \beta_{>0} \). \( \beta = \{\beta_0, \beta_{>0}\} \).

**Gaussian prior model:** For non-hierarchical models [12, 23], \( p_{\beta_{>0}}(z) \) is defined on single layer of latent variables and is typically assumed to be uniform or unit Gaussian. For hierarchical models with multiple layers of latent variables [32, 35], \( p_{\beta_{>0}}(z) \) can be further decomposed into conditional distributions between consecutive layers of latent variables as

\[
p_{\beta_{>0}}(z) = \prod_{i=1}^{L-1} p_{\beta_i}(z_i|z_{i+1})p(z_L)
\]

(2)

where \( p_{\beta_i}(z_i|z_{i+1}) \sim \mathcal{N}(\mu_{\beta_i}(z_{i+1}), \sigma^2_{\beta_i}(z_{i+1})) \) and is parameterized by a network with parameter \( \beta_i \), and \( p(z_L) \) is chosen to be a simple distribution, such as uniform or unit Gaussian.

**Maximum likelihood learning:** Learning such latent variable generative models can be done using maximum likelihood estimation (MLE). The marginal distribution is \( p_\beta(x) = \int p_\beta(x, z)dz \) with the gradient:

\[
\nabla_\beta \log p_\beta(x) = \mathbb{E}_{p_\beta(z|x)}[\nabla_\beta \log p_\beta(x, z)]
\]

(3)

where the expectation can be approximated via Monte Carlo sampling from the posterior distribution \( p_\beta(z|x) \). The MLE can then be accomplished through gradient ascent using such gradients. The posterior sampling usually requires the Markov Chain Monte Carlo (MCMC) such as Langevin dynamics [13, 32].

**Variational learning:** To alleviate the computational burden of MCMC, variational approach [4] introduces an additional inference model \( q_\omega(z|x) \) with a separate set of parameters \( \omega = (\omega_1, \ldots, \omega_L) \) for posterior approximation,

\[
q_\omega(z|x) = q_{\omega_1}(z_1|x) \prod_{i=1}^{L-1} q_{\omega_{i+1}}(z_{i+1}|z_i)
\]

(4)

where \( q_{\omega_1}(z_1|x) \) and \( q_{\omega_{i+1}}(z_{i+1}|z_i) \) are usually assumed as conditional Gaussian distributions, forming a “bottom-up” inference structure. The generator and inference model can be jointly learned via maximizing the evidence lower bound (ELBO), i.e., \( \max_{\beta, \omega} \text{ELBO}(\beta, \omega) \), where ELBO is defined as \( \text{ELBO}(\beta, \omega) = \mathbb{E}_{q_\omega(z|x)}[\log p_\beta(x|z)] - D_{\text{KL}}(q_\omega(z|x) || p_{\beta_{>0}}(z)) \).

2.2. Latent space energy-based model

The energy-based model (EBM) offers a flexible approach for learning the data distribution and is shown to be expressive in capturing data regularities [7, 8, 10, 31, 43, 45]. Most existing works focus on learning the EBM on data space, which is high-dimensional and can be challenging. To tackle this challenge, [2, 33] propose to learn latent space EBM as an informative prior model. With low-dimensional latent space, learning the EBM can be more efficient and effective, which in turn benefits the expressivity of the whole model. Specifically, [33] considers the latent space energy-based prior model on a single layer of latent variables,

\[
p_\alpha(z) = \frac{1}{Z(\alpha)} \exp [f_\alpha(z)] p_0(z)
\]

(5)

where \( -f_\alpha(z) \) is the energy function, \( Z(\alpha) \) is the normalizing constant, i.e., \( Z(\alpha) = \int \exp [f_\alpha(z)] p_0(z)dz \), and \( p_0(z) \) is the reference distribution assumed to be unit Gaussian. Compared to data space EBMs in which the energy function needs to support the entire high-dimensional space, such exponential tilting latent space EBMs can be more efficient in capturing data regularities.
3. Model and Learning

3.1. Joint latent space EBM prior model

For generator models with multi-layer latent variables (or multi-layer generator models), consecutive layers are modelled by conditional Gaussian distributions (see Eqn.2), which essentially assumes the conditional independence for latent units within the i-th layer given the (i + 1)-th layer of latent variables. Such a conditional independence assumption limits the model capacity as the contextual relation between latent units within each layer is largely ignored (see Fig.1), and needs to be improved for informative conditional modeling and better model expressivity. In this paper, we propose the joint EBM prior for multi-layer generator models,

\[ p_{\alpha,\beta > 0}(z) = \frac{1}{Z_{\alpha,\beta > 0}} \exp \left[ f_\alpha(z) \right] p_{\beta > 0}(z) \]

where we denote \( \alpha = (\alpha_1, ..., \alpha_L) \) for EBM parameters and \( z = (z_1, ..., z_L) \) for latent variables in different layers with layer \( L \) being the top layer. \( Z_{\alpha,\beta > 0} = \int \exp \left[ f_\alpha(z) \right] p_{\beta > 0}(z) dz \) is the normalizing constant regarding latent variables for all layers. Thus, the latent variables across different layers are jointly corrected via EBM prior as in Eqn.6, where \( f_\alpha(z) \) is the energy function for latent variables from all layers.

In this paper, we consider a simple factorized layer-wise parameterization, i.e., \( f_\alpha(z) = \sum_{i=1}^{L} f_{\alpha_i}(z_i) \), but other parameterizations are also feasible, which we will explore in future work. With such energy parameterization, it’s worth noting that the un-normalized prior model can be viewed as layer-wise exponential tilting,

\[
\exp \left[ \sum_{i=1}^{L} f_{\alpha_i}(z_i) \right] \prod_{i=1}^{L-1} p_{\beta_i}(z_i | z_{i+1}) p(z_L) \]

Energy correction

\[ = \frac{\exp \left[ f_{\alpha_L}(z_L) \right] p(z_L)}{\prod_{i=1}^{L-1} \exp \left[ f_\alpha(z_i) \right] p_\beta(z_i | z_{i+1})} \]

Correct on top layer

\[ \exp \left[ f_{\alpha}(z) \right] p(z) \prod_{i=1}^{L-1} \exp \left[ f_{\alpha_i}(z_i) \right] p_{\beta_i}(z_i | z_{i+1}) \]

Correct on intermediate layer

See Fig.1 for an illustration and comparison with multi-layer generator model with Gaussian prior.

**Joint vs. conditional EBM prior:** Besides the proposed joint modeling, it is also tempting to consider EBM prior for layer-wise Gaussian conditional, i.e., \( p_{\alpha_i,\beta_i}(z_i | z_{i+1}) = \frac{1}{Z(z_i, z_{i+1})} \exp \left[ f_{\alpha_i}(z_i) \right] p_{\beta_i}(z_i | z_{i+1}) \), and form the overall prior \( p_{\alpha,\beta > 0}(z) = \prod_{i=1}^{L} p_{\alpha_i,\beta_i}(z_i | z_{i+1}) p(z_L) \). Such a scheme is closely related to autoregressive energy machine and needs an additional inner-loop for sampling or optimization. The proposed joint EBM prior couples the latent variables across different layers via energy function and can be learned effectively and efficiently.

3.2. Maximum Likelihood Estimation

Our joint EBM prior model can be trained using MLE. Let \( \theta = (\alpha, \beta) \) denotes the model parameters and \( \theta \) can be learned by maximizing the log-likelihood on \( n \) training observations

\[
L(\theta) = \sum_{i=1}^{n} \log p_{\theta}(x_i) = \sum_{i=1}^{n} \log \int p_{\beta_i}(x | z)p_{\alpha,\beta > 0}(z)dz
\]

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

To update the parameter \( \theta \), we can compute the gradient of log-likelihood \( \nabla_{\theta} \log p_{\theta}(x) \) as

\[
\nabla_{\theta} \log p_{\theta}(x) = \mathbb{E}_{p_{\theta}(x,z)} \left[ \nabla_{\theta} \log p_{\theta}(x, z) \right]
\]

**Learning generation model** \( \beta_i \): \( p_{\beta_i}(x | z) \) is assumed to be Gaussian distribution, i.e., \( p_{\beta_i}(x | z) \sim \mathcal{N}(g_{\beta_i}(z), \sigma^2 I) \), with generation network \( g_{\beta_i} \) with parameter \( \beta_i \) and pre-specified \( \sigma^2 \) for simplicity. The learning gradient
\[ \nabla_{\beta_0} \log p_\theta(x) \text{ can then be expressed as} \]
\[ \nabla_{\beta_0} \log p_\theta(x) = E_{p_\theta(z|x)}[\nabla_{\beta_0} \log p_{\beta_0}(x|z)] \]
\[ = E_{p_\theta(z|x)} \left[ -\nabla_{\beta_0} \frac{||x - g_{\beta_0}(z)||^2}{2\sigma^2} \right] \]  

\textbf{Learning prior model } \alpha, \beta > 0: \text{ Learning } \alpha_i \text{ can be done by computing the gradient } \nabla_{\alpha_i} \log p_\theta(x) \text{ as} \]
\[ \nabla_{\alpha_i} \log p_\theta(x) = E_{p_\theta(z|x)}[\nabla_{\alpha_i} f_{\alpha_i}(z_i)] \]
\[ - E_{p_{\alpha,\beta > 0}} [\nabla_{\alpha_i} f_{\alpha_i}(z_i)] \]

\textbf{Sampling:} \text{ Both Eqn.10 and Eqn.11 require sampling from the posterior and prior distribution, which can be done via Langevin dynamic (LD) [26]. Given a target distribution } p(z), \text{ Langevin dynamic samples } z \sim p(z) \text{ by computing the gradient } \nabla_x \log p(z) \text{ and iteratively update } z \text{ as} \]
\[ z_t = z_{t-1} + s \nabla_x \log p(z_{t-1}) + \sqrt{2} \epsilon_{t-1} \]  

where \( t \) indexes the time step, \( s \) is the step size, and \( \epsilon \) is the Gaussian noise for each time step.

\textbf{Prior sampling:} \text{ By replacing target } p(z) \text{ with } p_{\alpha,\beta > 0}(z), \text{ the prior sampling computes } \nabla_x \log p_{\alpha,\beta > 0}(z) \text{ as} \]
\[ \nabla_x \left[ \sum_{i=1}^{L} f_{\alpha_i}(z_i) + \sum_{i=1}^{L-1} \log p_{\beta_i}(z_i|z_{i+1}) + \log p(z_L) \right] \]  

\textbf{Posterior sampling:} \text{ By replacing } p(z) \text{ with } p_\theta(z|x), \text{ where } p_\theta(z|x) \propto p_{\beta_0}(x|z)p_{\alpha,\beta > 0}(z), \text{ the posterior sampling computes } \nabla_x \log p_\theta(z|x) \text{ as} \]
\[ \nabla_x \log p_\theta(z|x) = \nabla_z [\log p_{\beta_0}(x|z) + \log p_{\alpha,\beta > 0}(z)] \]  

Notice that posterior sampling can be computationally inefficient as } \nabla_z [\log p_{\beta_0}(x|z)] \text{ requires back-propagation through the deep generation model.}

\textbf{3.3. Variational Learning} \]

For efficient posterior sampling, an inference model } q_\omega(z|x) \text{ with a separate set of parameters } \omega \text{ can be used for the posterior approximation. In this paper, we use the bottom-up inference model as Eqn.4 for amortizing the costly posterior MCMC sampling. Particularly, instead of KL minimization between marginal distributions as in MLE (see Sec.3.2), we consider the KL optimization between two joint densities, one for generator model density, i.e., } p_\theta(x, z) = p_{\beta_0}(x|z)p_{\alpha,\beta > 0}(z), \text{ and one for data density, i.e., } q_\omega(x, z) = p_{\text{data}}(x)q_\omega(z|x). \text{ We propose joint learning through KL minimization, denoting the objective to be } L(\theta, \omega), \text{i.e.,} \]
\[ \min_{\theta} \min_{\omega} L(\theta, \omega) = \min_{\theta} \min_{\omega} D_{KL}(q_\omega(x, z)||p_\theta(x, z)) \]  

\textbf{Learning generation model } \beta_0: \text{ For learning } \beta_0, \text{ we can compute the gradient as} \]
\[ -\nabla_{\beta_0} L(\theta, \omega) = \mathbb{E}_{p_{\text{data}}(x)}[E_{q_\omega(z|x)}[\nabla_{\beta_0} \log p_{\beta_0}(x|z)]] \]  

\textbf{Learning prior model } \alpha, \beta > 0: \text{ For learning } \alpha_i, \text{ the gradient is computed as} \]
\[ -\nabla_{\alpha_i} L(\theta, \omega) = \mathbb{E}_{p_{\text{data}}(x)}[\mathbb{E}_{q_\omega(z|x)}[\nabla_{\alpha_i} f_{\alpha_i}(z_i)]] 
- \mathbb{E}_{p_{\alpha,\beta > 0}} [\nabla_{\alpha_i} f_{\alpha_i}(z_i)] \]  

\textbf{Learning inference model } \omega: \text{ For learning } \omega_i, \text{ the gradient is} \]
\[ -\nabla_{\omega_i} L(\theta, \omega) = \nabla_{\omega_i} \mathbb{E}_{p_{\text{data}}(x)}[\mathbb{E}_{q_\omega(z|x)} \log p_{\beta_0}(x|z) 
- D_{KL}(q_\omega(z|x)||p_{\beta > 0}(z)) 
+ \mathbb{E}_{q_\omega(z|x)}[\sum_{i=1}^{L} f_{\alpha_i}(z_i)] \]  

\textbf{Divergence Perturbation.} \text{ The KL joint minimization (Eqn.15) can be viewed as a surrogate of the MLE objective with the KL perturbation term,} \]
\[ D_{KL}(q_\omega(z,x)||p_\theta(x,z)) = D_{KL}(p_{\text{data}}(x)||p_\theta(x,z)) + D_{KL}(q_\omega(z|x)||p_\theta(z|x)) \]  

where the perturbation term } D_{KL}(q_\omega(z|x)||p_\theta(z|x)) \text{ measures the KL-divergence between inference distribution and generator posterior. The inference model is learned to directly match the posterior distribution of the generator without expensive posterior sampling. In fact, such KL minimization in the joint space is closely related to evidence lower bound (ELBO) with the joint EBM as the prior model.

\textbf{4. Related Work} \]

\textbf{Hierarchical VAEs:} Variational auto-encoder (VAE) [23] proposes variational learning by introducing an approximation of the true intractable posterior, which allows a tractable bound on log-likelihood to be formed. But the non-hierarchical structure can be limited in model expressivity and fails to capture different levels of abstraction. Hierarchical VAEs (HVAEs) [5, 28, 35, 40] consist of multiple
layers of latent variables on top of the generator as a prior model, which can be used for learning complex data distribution and hierarchical representations. However, such models still focus on layer-wise relations while ignoring the intra-layer contextual relations at each layer.

**Energy-based models:** The energy-based models receive attention for being powerful and effective in capturing contextual data regularities. The majority of existing works focus on the pixel space \([7, 8, 10, 14, 15, 43–45]\). Learning such EBMs can be done using MLE, where MCMC sampling is typically required in each learning iteration which can be computationally expensive upon data space. Instead, \([33]\) proposes to build EBM on latent space where the energy function is considered as a correction of the non-informative Gaussian prior. The low dimensionality of the latent space makes EBM effective in capturing data regularities and can alleviate the burden of MCMC sampling.

**Generator models with informative prior:** For generator models, the assumed Gaussian or uniform prior distribution can be non-informative and less expressive. To address this problem, recent works \([2, 6, 11, 33, 38, 42]\) propose to learn generator models with an informative prior, where \(RAE\) \([11]\) constructs priors using rejection sampling, and Two-stage VAE \([6]\) propose to train an extra model for simple prior at the second stage to match the aggregated posterior distribution, while \(LEBM\) \([33]\) and \(NCP\)-VAE \([2]\) instead learn EBMs on latent space to improve the expressivity of generator models.

### 5. Experiments

To demonstrate the proposed method, we present extensive experiments, including (i) image synthesis, (ii) hierarchical representations, and (iii) analysis of latent space, as well as various ablation studies based on the proposed EBM prior. The parameter complexity is discussed in Sec.5.5.

#### 5.1. Image Synthesis

**Generator models with informative prior.** We first evaluate the generation performance of the proposed joint model. If the model is well-trained, the multi-layer EBM prior model should render an expressive prior distribution leading to realistic synthesis. We benchmark our model against other generator models that assume standard Gaussian prior, such as VAE \([23]\), Alternating Back-propagation (ABP) \([13]\), Ladder VAE (LVAE) \([35]\), and Short-run Inference (SRI) \([32]\), as well as other generator models using informative prior, such as \(RAE\) \([11]\), Two-stages VAE (2s-VAE) \([6]\), NCP-VAE \([2]\), and LEBM \([33]\), where LEBM builds EBM for single layer latent variables, while ours contains a multi-layer structure.

<table>
<thead>
<tr>
<th>Model</th>
<th>SVHN</th>
<th>CelebA-64</th>
<th>CIFAR-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAE ([23])</td>
<td>46.78</td>
<td>65.75</td>
<td>106.37</td>
</tr>
<tr>
<td>LVAE (L=5) ([35])</td>
<td>39.26</td>
<td>53.40</td>
<td>-</td>
</tr>
<tr>
<td>ABP ([13])</td>
<td>49.71</td>
<td>51.50</td>
<td>-</td>
</tr>
<tr>
<td>SRI (L=5) ([32])</td>
<td>35.32</td>
<td>47.95</td>
<td>-</td>
</tr>
<tr>
<td>RAE ([11])</td>
<td>42.02</td>
<td>40.95</td>
<td>74.16</td>
</tr>
<tr>
<td>2s-VAE ([6])</td>
<td>42.81</td>
<td>44.40</td>
<td>72.90</td>
</tr>
<tr>
<td>NCP-VAE ([2])</td>
<td>33.23</td>
<td>42.07</td>
<td>78.06</td>
</tr>
<tr>
<td>LEBM ([33])</td>
<td>29.44</td>
<td>37.87</td>
<td>70.15</td>
</tr>
<tr>
<td>Ours (L=2)</td>
<td>26.81</td>
<td>33.60</td>
<td>66.32</td>
</tr>
</tbody>
</table>

Table 1. FID(\(\downarrow\)) for our model and baselines on SVHN, CelebA (64 x 64), and CIFAR-10.

We train our model on SVHN \([29]\), CIFAR-10 \([24]\) and CelebA-64 \([27]\) and use Fréchet Inception Distance (FID) \([17]\) to quantitatively evaluate the generation quality. To make fair comparisons, we follow the standard protocol as in \([33]\) and use the same generation model with convolutional structures. We use Langevin posterior sampling for the training, and the generation model is jointly learned (the result for variational learning is shown in Ablation Studies). The comparisons are shown in Tab.1, where the superior generation performance indicates the effectiveness of our model in learning a more expressive prior.

**Toward deep hierarchical models.** We then consider the modern deep hierarchical structures as our multi-layer generator and explore the potential of the joint EBM prior for better generation. We adopt the two-stage training \([2, 43]\) where the deep multi-layer generator \(p_{\beta>0}(z)\) and inference model \(q_\omega(z|x)\) are trained in the first stage by maximizing the ELBO as in VAEs, with the pre-trained models, our joint EBM prior model can then be learned in the second stage where the posterior samples are directly obtained from the pre-trained inference model \(q_\omega(z|x)\) and prior samples can be obtained via Langevin sampling with change of variable on the generator \(p_{\beta>0}(z)\).

We consider NVAE \([40]\), a modern hierarchical VAE, for the first stage training, and we train our joint EBM prior in the second stage. For prior sampling in the second stage training, we employ similar reparameterized sampling scheme as in \([43]\) via provided code\(^1\) in order to better traverse the deep hierarchical latent space with different scales. We examine our model on CIFAR-10, CelebA-HQ-256 \([19]\), and LSUN-Church-64 \([46]\). The qualitative results for CelebA-HQ-256 and LSUN-Church-64 are shown in Fig.2 and Fig.3. For CelebA-HQ-256, we synthesize with adjusted batch-normalization as used in \([2, 43]\). We also visualize the Langevin transition on CIFAR-10 in Fig.4 where the quality of synthesis improves as the Langevin progresses.

\(^1\)https://github.com/NVlabs/VAEBM
Figure 2. Generated samples on CelebA-HQ-256. FID = 9.89.

Figure 3. Generated samples on LSUN-Church-64. FID = 8.38.

Figure 4. Langevin transitions on CIFAR-10. FID = 11.34.

Table 2. IS (↑) and FID (↓) for our model and baselines on CIFAR-10. Model* indicates backbone model.

<table>
<thead>
<tr>
<th>Method</th>
<th>IS</th>
<th>FID</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVAE* [40]</td>
<td>5.30</td>
<td>37.73</td>
</tr>
<tr>
<td>Ours</td>
<td>8.99</td>
<td>11.34</td>
</tr>
<tr>
<td>VAEBM [43]</td>
<td>8.43</td>
<td>12.19</td>
</tr>
<tr>
<td>Other EBMs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IGEbm [8]</td>
<td>6.78</td>
<td>38.2</td>
</tr>
<tr>
<td>ImprovedCD [7]</td>
<td>7.85</td>
<td>25.1</td>
</tr>
<tr>
<td>Divergence Triangle [14]</td>
<td>-</td>
<td>30.10</td>
</tr>
<tr>
<td>Adv-EBM [45]</td>
<td>9.10</td>
<td>13.21</td>
</tr>
<tr>
<td>Other Likelihood Models</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GLOW [22]</td>
<td>3.92</td>
<td>48.9</td>
</tr>
<tr>
<td>PixelCNN [41]</td>
<td>4.60</td>
<td>65.93</td>
</tr>
<tr>
<td>GANs+Score-based Models</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StyleGANv2 w/o ADA [20]</td>
<td>8.99</td>
<td>9.9</td>
</tr>
<tr>
<td>NCSN [36]</td>
<td>8.87</td>
<td>25.32</td>
</tr>
<tr>
<td>DDPM [18]</td>
<td>9.46</td>
<td>3.17</td>
</tr>
</tbody>
</table>

Table 3. FID (↓) for our model and baselines on CelebA-HQ-256 and LSUN-Church-64. Model* indicates backbone model.

<table>
<thead>
<tr>
<th>Model</th>
<th>CelebA-HQ-256</th>
<th>LSUN-Church-64</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVAE* [40]</td>
<td>30.25</td>
<td>38.13</td>
</tr>
<tr>
<td>Ours</td>
<td>9.89</td>
<td>8.38</td>
</tr>
<tr>
<td>NCP-VAE [2]</td>
<td>24.79</td>
<td>-</td>
</tr>
<tr>
<td>VAEBM [43]</td>
<td>20.38</td>
<td>13.51</td>
</tr>
<tr>
<td>Adv-EBM [45]</td>
<td>17.31</td>
<td>10.84</td>
</tr>
<tr>
<td>GLOW [22]</td>
<td>68.93</td>
<td>59.35</td>
</tr>
<tr>
<td>PGGAN [19]</td>
<td>8.03</td>
<td>6.42</td>
</tr>
</tbody>
</table>

5.2. Hierarchical Representations

Hierarchical sampling. To examine our model in learning hierarchical representation, we employ hierarchical sampling to illustrate the learned representation at different layers. In particular, we first sample one group of latent vectors from EBM prior and hold them as fixed constants, then we randomly sample multiple groups of latent vectors to replace the fixed latent vectors at different layers. This allows us to visualize the variation in representation across layers.

We apply our joint EBM prior to BIVA [28] on CelebA-64. For training, we reuse the two-stage training scheme, where we recruit BIVA\(^2\) for the first stage training and train our EBM prior model in the second stage by using the reparametrized sampling [43] method (similar as deep hierarchical models in Sec. 5.1). We show the results of hierarchical sampling in Fig. 6 and observe that BIVA presents minor changes at the bottom and middle layers, while the proposed joint EBM prior model can show variations of different levels. Note that it is a challenging task for conditional hierarchical models [48], on which the improvement thus suggests that the proposed method is capable of learning hierarchical representations for multi-layer generator models.

\(^2\)https://github.com/vlievin/biva-pytorch
Out-of-distribution detection. Next, we conduct out-of-distribution (OOD) detection to further evaluate the hierarchical representations. Typically, low-level representations (e.g., edges, corners) can be shared across data which in turn leads to high-confidence reconstructions for OOD examples, while high-level semantic ones have fewer correlations across different data and shall be more discriminative for OOD detection. Inspired by [16], we consider an unnormalized log-posterior as the decision function for EBM prior model, which is defined as

\[
L_{\text{EBM}}^{\geq k} = \mathbb{E}_{\mathbf{z} \sim q_\omega(x), \mathbf{z} \leq k \sim p_{\beta_0, \alpha}(\mathbf{z})} \left[ \log p_{\beta_0}(x|\mathbf{z}) + \log p_{\beta_0}(\mathbf{z}) + \sum_{i=1}^{L} f_{\alpha_i}(\mathbf{z}_i) \right]
\]

(20)

where latent codes above the \( k \)-th layer are inferred from inference model and kept fixed, and those below the \( k \)-th layer are sampled from EBM prior via the reparameterized sampling \(^3\) [43] with fixed inferred latent codes. With \( k = 0 \), all layers of latent vectors are inferred from \( q_\omega(x|\mathbf{z}) \). With a higher value of \( k \), less inferred low-level representations are used, which should render better performance in OOD detection. In addition, we can also compute a subtraction between \( L_{\text{EBM}}^{\geq 0} \) and \( L_{\text{EBM}}^{\geq k} \) as a surrogate of the likelihood-ratio which is shown to be effective for OOD detection [16]. We compute the subtraction as

\[
LLR_{\text{EBM}}^{\geq k} = L_{\text{EBM}}^{\geq 0} - L_{\text{EBM}}^{\geq k}
\]

(21)

We follow standard protocols and apply our EBM prior model with BIVA on CIFAR-10 and use SVHN as OOD data for testing. In Fig.5, we show the density of in-distribution and OOD data by computing the unnormalized log-posterior with increased \( k \), and we use AUROC, AUPRC to quantitatively evaluate the performance. It can be seen that as \( k \) increases, relatively lower log-likelihoods are assigned to OOD data, which in turn renders better detection performance (higher AUROC and AUPRC). More importantly, we observe that the backbone model BIVA achieves the best detection performance of 0.885 for AUROC, while our models achieve 0.927 with the adapted decision function. This further verifies that the hierarchical representations can be learned within our multi-layer structure.

5.3. Analysis of Latent Space

Long-run langevin transition. In this section, we examine the energy landscape of our joint EBM prior model. If the EBM is well-learned, the energy prior should naturally render local modes of the energy function, and traversing these local modes should present realistic synthesized examples and steady-state energy scores. Existing EBMs typically have oversaturated images via long-run Langevin dynamics as observed in [30]. Training an EBM that learns steady-state energy scores over realistic images can be useful but challenging.

We train our model on CelebA-64 using Langevin dynamics for 40 steps. We then run 100 and 2500 Langevin steps to examine the learned energy landscape. We show the synthesis and corresponding energy profile in Fig.7. It can be seen that generated examples become sharper for the first 40 steps as it starts from the referenced distribution \( p_{\beta_0}(\mathbf{z}) \) toward the learned energy prior \( p_{\beta_0, \alpha}(\mathbf{z}) \), and the energy fluctuates around some constant. For long-run 2500 steps, it is worth noting that our EBM prior model delivers diverse and realistic synthesis, and it does not exhibit the oversaturated phenomenon. This suggests that the learned EBM could mix well between different local modes of the learned energy prior.

\(^3\)https://github.com/NVlabs/VAEBM
Anomaly Detection. We further evaluate how our joint EBM prior model could benefit the anomaly detection (AD) task. Different from OOD detection, AD requires one class (e.g., one-digit class from MNIST) of data to be held out as anomaly for training, and both normal (e.g., other nine-digit classes from MNIST) and anomalous data are used for testing.

The proposed prior model is built on the joint of all layers of latent variables. If it is well learned, the posterior \( q_z(x) \) could form a discriminative joint latent space that has separated probability densities for normal and anomalous data. We use un-normalized log-posterior \( \log p(x) \) as our decision function and train our model on MNIST with each class held out as an anomalous class. We consider the baseline models that also adopt an inferential mechanism, such as VAE [23], MEG [25], BiGAN-\( \sigma \) [47], OT-SRI [1], and LEBM [33] which assumes single-layer latent space and is closely related to our method. Tab.4 shows the results of AUPRC scores averaged over the last 10 epochs to account for the variance. To make fair comparisons, we follow the protocols in [1, 25, 33, 47].

Table 4. AUPRC scores for unsupervised anomaly detection.

<table>
<thead>
<tr>
<th>Method</th>
<th>1</th>
<th>5</th>
<th>25</th>
<th>50</th>
<th>75</th>
<th>90</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAE [23]</td>
<td>0.063</td>
<td>0.137</td>
<td>0.325</td>
<td>0.448</td>
<td>0.594</td>
<td></td>
</tr>
<tr>
<td>MEG [25]</td>
<td>0.281±0.035</td>
<td>0.466±0.066</td>
<td>0.428±0.068</td>
<td>0.290±0.040</td>
<td>0.342±0.034</td>
<td>0.567±0.038</td>
</tr>
<tr>
<td>BiGAN-( \sigma ) [47]</td>
<td>0.287±0.023</td>
<td>0.443±0.029</td>
<td>0.534±0.029</td>
<td>0.547±0.017</td>
<td>0.367±0.028</td>
<td>0.555±0.023</td>
</tr>
<tr>
<td>OT-SRI [1]</td>
<td>0.336±0.021</td>
<td>0.775±0.024</td>
<td>0.708±0.030</td>
<td>0.590±0.018</td>
<td>0.555±0.018</td>
<td>0.555±0.023</td>
</tr>
<tr>
<td>LEBM [33]</td>
<td>0.336±0.008</td>
<td>0.690±0.017</td>
<td>0.619±0.013</td>
<td>0.683±0.009</td>
<td>0.413±0.010</td>
<td>0.413±0.010</td>
</tr>
<tr>
<td>Ours</td>
<td>0.470±0.009</td>
<td>0.941±0.001</td>
<td>0.964±0.003</td>
<td>0.815±0.004</td>
<td>0.756±0.004</td>
<td>0.756±0.004</td>
</tr>
</tbody>
</table>

Table 4. AUPRC scores for unsupervised anomaly detection.

5.4. Ablation Studies.

MCMC sampling vs. Inference model. Two posterior sampling schemes using MCMC and inference model are compared in Tab.5 in terms of FID and wall-clock training time (per-iteration). The MCMC posterior sampling renders better FID as it is more accurate in inference [13, 32], but it can be computationally heavy. While the inference model is efficient in learning but can be less accurate. For deep hierarchical structures, the variational learning with inference model is preferred due to its efficiency.

Table 5. FID and training time for MCMC posterior sampling and variational learning.

<table>
<thead>
<tr>
<th>Method</th>
<th>SVHN</th>
<th>CelebA-64</th>
<th>CIFAR-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCMC / Inf</td>
<td>26.81 / 28.60</td>
<td>33.60 / 36.12</td>
<td>66.32 / 68.45</td>
</tr>
</tbody>
</table>

Langevin steps. We explore the different number of Langevin steps in prior sampling for training on CIFAR-10. The results of FID and corresponding training time are shown in Tab.6. We observe that the Langevin step \( k \) increasing from 10 to 40 can improve the generation quality, while for steps more than 40, it only has minor impacts on the improvement but with increased training overhead. We thus report the result of \( k = 40 \) in Tab.1.

Table 6. FID and training time for increasing MCMC steps in prior sampling.

<table>
<thead>
<tr>
<th>( k )</th>
<th>FID</th>
<th>Time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>69.42 / 67.58</td>
<td>0.312 / 0.232</td>
</tr>
<tr>
<td>20</td>
<td>69.39 / 67.56</td>
<td>0.349 / 0.246</td>
</tr>
<tr>
<td>40</td>
<td>66.32 / 68.45</td>
<td>0.478 / 0.337</td>
</tr>
</tbody>
</table>

Table 7. FID and parameter complexity for backbone model and EBM.

<table>
<thead>
<tr>
<th>Method</th>
<th>CIFAR-10</th>
<th>CelebA-HQ-256</th>
<th>LSUN-Church-64</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVAE / EBM</td>
<td>59.73 / 11.34</td>
<td>30.25 / 9.89</td>
<td>38.13 / 8.38</td>
</tr>
</tbody>
</table>

| Parameters  | 257M / 9M (3%) | 375M / 18M (4%) | 65M / 5M (7%) |

Table 7. FID and parameter complexity for backbone model and EBM.

6. Conclusion

we propose a joint EBM prior for multi-layer generator models, which can effectively capture the intra-layer relations at each layer and jointly correct the latent variables from all layers. We present a joint training scheme via MLE and further develop a variational learning scheme for efficient inference. Our comprehensive experiments demonstrate the effectiveness of the proposed method.
References


[28] Lars Maaløe, Marco Fraccaro, Valentín Liévin, and Ole Winther. Biva: A very deep hierarchy of latent variables for


