

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

Jiali Cui¹, Ying Nian Wu², Tian Han^{1*}

¹Department of Computer Science, Stevens Institute of Technology

²Department of Statistics, University of California, Los Angeles

{jcui7, 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 [22, 35, 38, 40] as well as representation learning [5, 49]. 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, 29, 33, 36, 41], 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, [34] 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 [34] 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

*: corresponding author

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

$$p_\beta(\mathbf{x}, \mathbf{z}) = p_{\beta_0}(\mathbf{x}|\mathbf{z})p_{\beta_{>0}}(\mathbf{z}) \quad (1)$$

where $p_{\beta_0}(\mathbf{x}|\mathbf{z})$ is the generation model with parameter β_0 that maps from latent space to data space, and $p_{\beta_{>0}}(\mathbf{z})$ is the prior distribution over latent variables with parameter $\beta_{>0}$. $\beta = \{\beta_0, \beta_{>0}\}$.

Gaussian prior model: For non-hierarchical models [13, 24], $p_{\beta_{>0}}(\mathbf{z})$ 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 [33, 36], $p_{\beta_{>0}}(\mathbf{z})$ can be further decomposed into conditional distributions between consecutive layers of latent variables as

$$p_{\beta_{>0}}(\mathbf{z}) = \prod_{i=1}^{L-1} p_{\beta_i}(\mathbf{z}_i|\mathbf{z}_{i+1})p(\mathbf{z}_L) \quad (2)$$

where $p_{\beta_i}(\mathbf{z}_i|\mathbf{z}_{i+1}) \sim \mathcal{N}(\mu_{\beta_i}(\mathbf{z}_{i+1}), \sigma_{\beta_i}^2(\mathbf{z}_{i+1}))$ and is parameterized by a network with parameter β_i , and $p(\mathbf{z}_L)$ 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 $p_\beta(\mathbf{x}) = \int p_\beta(\mathbf{x}, \mathbf{z})d\mathbf{z}$ with the gradient:

$$\nabla_\beta \log p_\beta(\mathbf{x}) = \mathbb{E}_{p_\beta(\mathbf{z}|\mathbf{x})}[\nabla_\beta \log p_\beta(\mathbf{x}, \mathbf{z})] \quad (3)$$

where the expectation can be approximated via Monte Carlo sampling from the posterior distribution $p_\beta(\mathbf{z}|\mathbf{x})$. 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 [14, 33].

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

$$q_\omega(\mathbf{z}|\mathbf{x}) = q_{\omega_1}(\mathbf{z}_1|\mathbf{x}) \prod_{i=1}^{L-1} q_{\omega_{i+1}}(\mathbf{z}_{i+1}|\mathbf{z}_i) \quad (4)$$

where $q_{\omega_1}(\mathbf{z}_1|\mathbf{x})$ and $q_{\omega_{i+1}}(\mathbf{z}_{i+1}|\mathbf{z}_i)$ 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 $\text{ELBO}(\beta, \omega) = \mathbb{E}_{q_\omega(\mathbf{z}|\mathbf{x})}[\log p_{\beta_0}(\mathbf{x}|\mathbf{z})] - D_{\text{KL}}(q_\omega(\mathbf{z}|\mathbf{x})||p_{\beta_{>0}}(\mathbf{z}))$.

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 [8, 9, 11, 32, 44, 46]. Most existing works focus on learning the EBM on data space, which is high-dimensional and can be challenging. To tackle this challenge, [2, 6, 34] 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, [34] considers the latent space energy-based prior model on a *single layer* of latent variables,

$$p_\alpha(\mathbf{z}) = \frac{1}{Z(\alpha)} \exp[-f_\alpha(\mathbf{z})]p_0(\mathbf{z}) \quad (5)$$

where $-f_\alpha(\mathbf{z})$ is the energy function, $Z(\alpha)$ is the normalizing constant, i.e., $Z(\alpha) = \int \exp[-f_\alpha(\mathbf{z})]p_0(\mathbf{z})d\mathbf{z}$, and $p_0(\mathbf{z})$ 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 model*), 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,

$$p_{\alpha, \beta_{>0}}(\mathbf{z}) = \frac{1}{Z_{\alpha, \beta_{>0}}} \exp [f_{\alpha}(\mathbf{z})] p_{\beta_{>0}}(\mathbf{z}) \quad (6)$$

$$= \frac{1}{Z_{\alpha, \beta_{>0}}} \exp \left[\sum_{i=1}^L f_{\alpha_i}(\mathbf{z}_i) \right] \prod_{i=1}^{L-1} p_{\beta_i}(\mathbf{z}_i | \mathbf{z}_{i+1}) p(\mathbf{z}_L)$$

where we denote $\alpha = (\alpha_1, \dots, \alpha_L)$ for EBM parameters and $\mathbf{z} = (\mathbf{z}_1, \dots, \mathbf{z}_L)$ for latent variables in different layers with layer L being the top layer. $Z_{\alpha, \beta_{>0}} = \int \exp [f_{\alpha}(\mathbf{z})] p_{\beta_{>0}}(\mathbf{z}) d\mathbf{z}$ 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 $f_{\alpha}(\mathbf{z})$ is the energy function for latent variables from all layers.

In this paper, we consider a simple factorized layer-wise parameterization, i.e., $f_{\alpha}(\mathbf{z}) = \sum_{i=1}^L f_{\alpha_i}(\mathbf{z}_i)$, 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,

$$\underbrace{\exp \left[\sum_{i=1}^L f_{\alpha_i}(\mathbf{z}_i) \right]}_{\text{Energy correction}} \underbrace{\prod_{i=1}^{L-1} p_{\beta_i}(\mathbf{z}_i | \mathbf{z}_{i+1}) p(\mathbf{z}_L)}_{\text{Gaussian prior}} \quad (7)$$

$$= \underbrace{\exp [f_{\alpha_L}(\mathbf{z}_L)] p(\mathbf{z}_L)}_{\text{Correction on top layer}} \prod_{i=1}^{L-1} \underbrace{\exp [f_{\alpha_i}(\mathbf{z}_i)] p_{\beta_i}(\mathbf{z}_i | \mathbf{z}_{i+1})}_{\text{Correction on intermediate layer}}$$

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., $\tilde{p}_{\alpha_i, \beta_i}(\mathbf{z}_i | \mathbf{z}_{i+1}) = \frac{1}{Z(\mathbf{z}_{i+1})} \exp [f_{\alpha_i}(\mathbf{z}_i)] p_{\beta_i}(\mathbf{z}_i | \mathbf{z}_{i+1})$, and form the overall prior $p_{\alpha, \beta_{>0}}(\mathbf{z}) = \prod_{i=1}^L \tilde{p}_{\alpha_i, \beta_i}(\mathbf{z}_i | \mathbf{z}_{i+1}) p(\mathbf{z}_L)$. Such a scheme is closely related to autoregressive energy machine [10] and is adopted in NCP-VAE [2]. However, the normalizing constant $Z(\mathbf{z}_{i+1})$ in $\tilde{p}_{\alpha_i, \beta_i}(\mathbf{z}_i | \mathbf{z}_{i+1})$ involves the

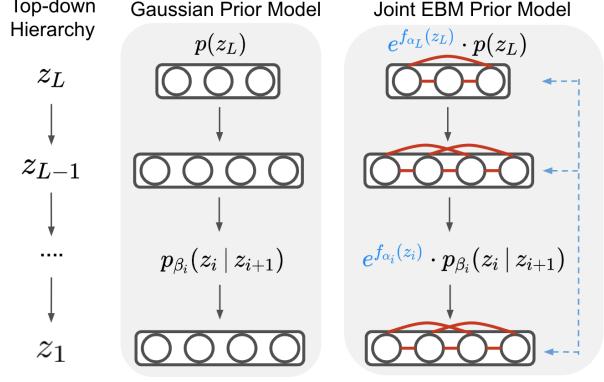


Figure 1. **Left panel:** Gaussian prior model. **Right panel:** Joint EBM prior model. **Black solid lines with arrow:** inter-layer relations modelling. **Red solid lines:** intra-layer contextual relations modelling. **Blue dashed lines:** joint modelling upon all layers.

latent variable \mathbf{z}_{i+1} from the upper layer which can be *intractable* 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 θ can be learned by maximizing the log-likelihood on n training observations

$$L(\theta) = \sum_{i=1}^n \log p_{\theta}(\mathbf{x}_i) = \sum_{i=1}^n \log \int p_{\beta_0}(\mathbf{x} | \mathbf{z}) p_{\alpha, \beta_{>0}}(\mathbf{z}) d\mathbf{z}$$

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., $\min_{\theta} D_{\text{KL}}(p_{\text{data}}(\mathbf{x}) || p_{\theta}(\mathbf{x}))$.

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

$$\begin{aligned} \nabla_{\theta} \log p_{\theta}(\mathbf{x}) &= \mathbb{E}_{p_{\theta}(\mathbf{z} | \mathbf{x})} [\nabla_{\theta} \log p_{\theta}(\mathbf{x}, \mathbf{z})] \\ &= \mathbb{E}_{p_{\theta}(\mathbf{z} | \mathbf{x})} [\nabla_{\theta} \log p_{\beta_0}(\mathbf{x} | \mathbf{z})] \\ &\quad + \mathbb{E}_{p_{\theta}(\mathbf{z} | \mathbf{x})} [\nabla_{\theta} \log p_{\alpha, \beta_{>0}}(\mathbf{z})] \end{aligned} \quad (8)$$

With such a gradient, we can learn θ using gradient ascent. **Learning generation model β_0 :** $p_{\beta_0}(\mathbf{x} | \mathbf{z})$ is assumed to be Gaussian distribution, i.e., $p_{\beta_0}(\mathbf{x} | \mathbf{z}) \sim \mathcal{N}(g_{\beta_0}(\mathbf{z}), \sigma^2 I)$, with generation network g_{β_0} with parameter β_0 and pre-specified σ^2 for simplicity. The learning gradient $\nabla_{\beta_0} \log p_{\theta}(\mathbf{x})$ can then be expressed as

$$\begin{aligned} \nabla_{\beta_0} \log p_{\theta}(\mathbf{x}) &= \mathbb{E}_{p_{\theta}(\mathbf{z} | \mathbf{x})} [\nabla_{\beta_0} \log p_{\beta_0}(\mathbf{x} | \mathbf{z})] \\ &= \mathbb{E}_{p_{\theta}(\mathbf{z} | \mathbf{x})} \left[-\nabla_{\beta_0} \frac{\|\mathbf{x} - g_{\beta_0}(\mathbf{z})\|^2}{2\sigma^2} \right] \end{aligned} \quad (9)$$

Learning prior model $\alpha, \beta_{>0}$: Learning α_i can be done by computing the gradient $\nabla_{\alpha_i} \log p_\theta(\mathbf{x})$ as

$$\begin{aligned}\nabla_{\alpha_i} \log p_\theta(\mathbf{x}) &= \mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{x})}[\nabla_{\alpha_i} f_{\alpha_i}(\mathbf{z}_i)] \quad (10) \\ &- \mathbb{E}_{p_{\alpha, \beta_{>0}}(\mathbf{z})}[\nabla_{\alpha_i} f_{\alpha_i}(\mathbf{z}_i)]\end{aligned}$$

For updating $\beta_{>0}$, the gradient $\nabla_{\beta_i} \log p_\theta(\mathbf{x})$ is

$$\begin{aligned}\nabla_{\beta_i} \log p_\theta(\mathbf{x}) &= \mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{x})}[\nabla_{\beta_i} \log p_{\beta_i}(\mathbf{z}_i|\mathbf{z}_{i+1})] \quad (11) \\ &- \mathbb{E}_{p_{\alpha, \beta_{>0}}(\mathbf{z})}[\nabla_{\beta_i} \log p_{\beta_i}(\mathbf{z}_i|\mathbf{z}_{i+1})]\end{aligned}$$

Sampling: Both Eqn.10 and Eqn.11 require sampling from the posterior and prior distribution, which can be done via Langevin dynamic (LD) [27]. Given a target distribution $p(\mathbf{z})$, Langevin dynamic samples $\mathbf{z} \sim p(\mathbf{z})$ by computing the gradient $\nabla_{\mathbf{z}} \log p(\mathbf{z})$ and iteratively update \mathbf{z} as

$$\mathbf{z}_t = \mathbf{z}_{t-1} + s \nabla_{\mathbf{z}} \log p(\mathbf{z}_{t-1}) + \sqrt{2s} \epsilon_{t-1} \quad (12)$$

where t indexes the time step, s is the step size, and ϵ is the Gaussian noise for each time step.

Prior sampling: By replacing target $p(\mathbf{z})$ with $p_{\alpha, \beta_{>0}}(\mathbf{z})$, the prior sampling computes $\nabla_{\mathbf{z}} \log p_{\alpha, \beta_{>0}}(\mathbf{z})$ as

$$\nabla_{\mathbf{z}} \left[\sum_{i=1}^L f_{\alpha_i}(\mathbf{z}_i) + \sum_{i=1}^{L-1} \log p_{\beta_i}(\mathbf{z}_i|\mathbf{z}_{i+1}) + \log p(\mathbf{z}_L) \right] \quad (13)$$

Posterior sampling: By replacing $p(\mathbf{z})$ with $p_\theta(\mathbf{z}|\mathbf{x})$, where $p_\theta(\mathbf{z}|\mathbf{x}) \propto p_{\beta_0}(\mathbf{x}|\mathbf{z}) p_{\alpha, \beta_{>0}}(\mathbf{z})$, the posterior sampling computes $\nabla_{\mathbf{z}} \log p_\theta(\mathbf{z}|\mathbf{x})$ as

$$\nabla_{\mathbf{z}} \log p_\theta(\mathbf{z}|\mathbf{x}) = \nabla_{\mathbf{z}} [\log p_{\beta_0}(\mathbf{x}|\mathbf{z}) + \log p_{\alpha, \beta_{>0}}(\mathbf{z})] \quad (14)$$

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

3.3. Variational Learning

For efficient posterior sampling, an inference model $q_\omega(\mathbf{z}|\mathbf{x})$ with a separate set of parameters ω 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., $p_\theta(\mathbf{x}, \mathbf{z}) = p_{\beta_0}(\mathbf{x}|\mathbf{z}) p_{\alpha, \beta_{>0}}(\mathbf{z})$, and one for data density, i.e., $q_\omega(\mathbf{x}, \mathbf{z}) = p_{\text{data}}(\mathbf{x}) q_\omega(\mathbf{z}|\mathbf{x})$. We propose joint learning through KL minimization, denoting the objective to be $L(\theta, \omega)$, i.e.,

$$\min_{\theta} \min_{\omega} L(\theta, \omega) = \min_{\theta} \min_{\omega} D_{\text{KL}}(q_\omega(\mathbf{x}, \mathbf{z}) || p_\theta(\mathbf{x}, \mathbf{z})) \quad (15)$$

Learning generation model β_0 : For learning β_0 , we can compute the gradient as

$$-\nabla_{\beta_0} L(\theta, \omega) = \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \mathbb{E}_{q_\omega(\mathbf{z}|\mathbf{x})} [\nabla_{\beta_0} \log p_{\beta_0}(\mathbf{x}|\mathbf{z})] \quad (16)$$

Learning prior model $\alpha, \beta_{>0}$: For learning α_i , the gradient is computed as

$$\begin{aligned}-\nabla_{\alpha_i} L(\theta, \omega) &= \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \mathbb{E}_{q_\omega(\mathbf{z}|\mathbf{x})} [\nabla_{\alpha_i} f_{\alpha_i}(\mathbf{z}_i)] \\ &- \mathbb{E}_{p_{\alpha, \beta_{>0}}(\mathbf{z})} [\nabla_{\alpha_i} f_{\alpha_i}(\mathbf{z}_i)]\end{aligned} \quad (17)$$

For learning $\beta_{>0}$, we compute the gradient as

$$\begin{aligned}-\nabla_{\beta_i} L(\theta, \omega) &= \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \mathbb{E}_{q_\omega(\mathbf{z}|\mathbf{x})} [\nabla_{\beta_i} \log p_{\beta_i}(\mathbf{z}_i|\mathbf{z}_{i+1})] \\ &- \mathbb{E}_{p_{\alpha, \beta_{>0}}(\mathbf{z})} [\nabla_{\beta_i} \log p_{\beta_i}(\mathbf{z}_i|\mathbf{z}_{i+1})]\end{aligned} \quad (18)$$

Learning inference model ω : For learning ω_i , the gradient is

$$\begin{aligned}-\nabla_{\omega_i} L(\theta, \omega) &= \nabla_{\omega_i} \mathbb{E}_{p_{\text{data}}(\mathbf{x})} [\mathbb{E}_{q_\omega(\mathbf{z}|\mathbf{x})} \log p_{\beta_0}(\mathbf{x}|\mathbf{z})] \\ &- D_{\text{KL}}(q_\omega(\mathbf{z}|\mathbf{x}) || p_{\beta_{>0}}(\mathbf{z})) \\ &+ \mathbb{E}_{q_\omega(\mathbf{z}|\mathbf{x})} [\sum_{i=1}^L f_{\alpha_i}(\mathbf{z}_i)]\end{aligned} \quad (19)$$

We refer to detailed derivation in Appendix A.

Divergence Perturbation. The KL joint minimization (Eqn.15) can be viewed as a surrogate of the MLE objective with the KL perturbation term,

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

where the perturbation term $D_{\text{KL}}(q_\omega(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z}|\mathbf{x}))$ 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.

4. Related Work

Hierarchical VAEs: Variational auto-encoder (VAE) [24] 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, 29, 36, 41] 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 expressive and powerful in capturing contextual data regularities. The majority of existing works focus on the pixel space [8, 9, 11, 15, 16, 44–46]. 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, [34] 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, 7, 12, 34, 39, 43] propose to learn generator models with an informative prior, where RAE [12] constructs priors using rejection sampling, and Two-stage VAE [7] propose to train an extra model for simple prior at the second stage to match the aggregated posterior distribution, while LEBM [34] 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) latent visualization, (ii) image synthesis, (iii) hierarchical representations, and (iv) analysis of latent space. To better understand the proposed model, we conduct various ablation studies based on the proposed EBM prior in Sec.5.5. The parameter complexity is discussed in Sec.5.6.

5.1. Latent Visualization

We examine the expressivity of our EBM prior model by latent visualization. We pick MNIST data with only digit classes ‘1’ and ‘0’ available, on which we train our 2-layer model with the latent dimension of each layer set to be 2 for better visualization. We train with $k = 40$ steps for prior sampling and visualize the transition of Langevin dynamics on each layer for every 10 steps in Fig.2. It can be seen that the latent variables are first initialized from Gaussian noise and then can be tilted to match the multi-modal posterior, for which the standard Gaussian prior can be infeasible.

5.2. Image Synthesis

Generator models with informative prior. We 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 [24], Alternating Back-propagation (ABP) [14], Ladder VAE (LVAE) [36], and Short-run Inference (SRI)

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

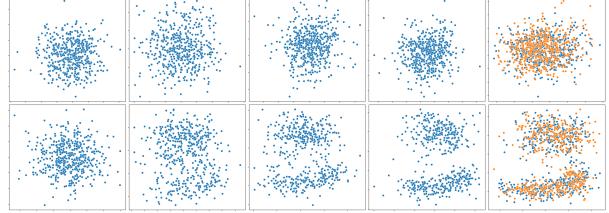


Figure 2. Langevin transition on latent codes (bottom: z_1 , top: z_2). **Blue** color indicates the transition of Langevin prior sampling. **Orange** color indicates latent codes inferred from inference model.

[33], as well as other generator models using informative prior, such as RAE [12], Two-stages VAE (2s-VAE) [7], NCP-VAE [2], and LEBM [34], where LEBM builds EBM for single layer latent variables, while ours contains a multi-layer structure.

We train our model on SVHN [30], CIFAR-10 [25] and CelebA-64 [28] and use Fréchet Inception Distance (FID) [18] to quantitatively evaluate the generation quality. To make fair comparisons, we follow the standard protocol as in [34] 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.

Model	SVHN	CelebA-64	CIFAR-10
VAE [24]	46.78	65.75	106.37
LVAE (L=5) [36]	39.26	53.40	-
ABP [14]	49.71	51.50	-
SRI (L=5) [33]	35.32	47.95	-
RAE [12]	42.02	40.95	74.16
2s-VAE [7]	42.81	44.40	72.90
NCP-VAE [2]	33.23	42.07	78.06
LEBM [34]	29.44	37.87	70.15
Ours (L=2)	26.81	33.60	66.32

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

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, 44] where the deep multi-layer generator $p_{\beta>0}(\mathbf{z})$ and inference model $q_{\omega}(\mathbf{z}|\mathbf{x})$ 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



Figure 3. Generated samples on CelebA-HQ-256 . FID = 9.89.



Figure 4. Generated samples on LSUN-Church-64. FID = 8.38.

pre-trained inference model $q_\omega(\mathbf{z}|\mathbf{x})$ and prior samples can be obtained via Langevin sampling with change of variable on the generator $p_{\beta>0}(\mathbf{z})$ (see details in Appendix.A.3).

We consider NVAE [41], 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 *reparametrized sampling* scheme as in [44] via provided code¹ in order to better traverse the deep hierarchical latent space with different scales. We examine our model on CIFAR-10, CelebA-HQ-256 [20], and LSUN-Church-64 [47]. The qualitative results for CelebA-HQ-256 and LSUN-Church-64 are shown in Fig.3 and Fig.4. For CelebA-HQ-256, we synthesize with adjusted batch-normalization as used in [2, 44]. We also visualize the Langevin transition on CIFAR-10 in Fig.5 where the quality of synthesis improves as the Langevin progresses. We refer to more results in Appendix.D

The quantitative results are shown in Tab.2 and Tab.3. We consider the baseline models, including NCP-VAE [2] and VAEBM [44], which also recruit NVAE as their backbone model, and other powerful deep generative models, such as GANs [3, 21], score-based models [19, 37] and EBMs [8, 9, 15, 46] on data space. Compared to NVAE backbone model, our joint EBM prior model can significantly improve the fidelity of generated samples while only accounting for negligible overhead (see Parameter Efficiency in Sec.5.6). In comparison with other powerful deep generative models, we also achieve competitive generation performance.

5.3. 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



Figure 5. Langevin transitions on CIFAR-10. FID = 11.34.

Method	IS	FID
NVAE* [41]	5.30	37.73
Ours	8.99	11.34
NCP-VAE [2]	-	24.08
VAEBM [44]	8.43	12.19
Other EBMs		
IGEBM [9]	6.78	38.2
ImprovedCD [8]	7.85	25.1
Divergence Triangle [15]	-	30.10
Adv-EBM [46]	9.10	13.21
Other Likelihood Models		
GLOW [23]	3.92	48.9
PixelCNN [42]	4.60	65.93
GANs+Score-based Models		
BigGAN [3]	9.22	14.73
StyleGANv2 w/o ADA [21]	8.99	9.9
NCSN [37]	8.87	25.32
DDPM [19]	9.46	3.17

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

Model	CelebA-HQ-256	LSUN-Church-64
NVAE* [41]	30.25	38.13
Ours	9.89	8.38
NCP-VAE [2]	24.79	-
VAEBM [44]	20.38	13.51
Adv-EBM [46]	17.31	10.84
GLOW [23]	68.93	59.35
PGGAN [20]	8.03	6.42

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

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.

¹<https://github.com/NVlabs/VAEBM>

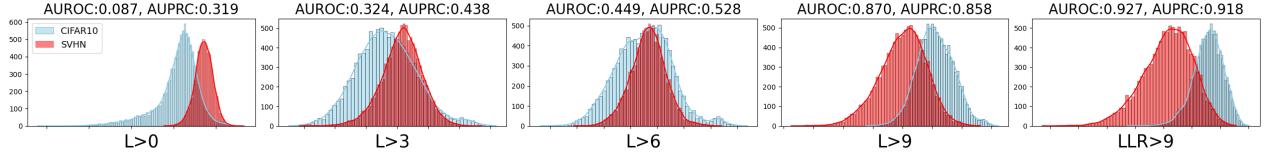


Figure 6. Histograms of density of $L_{\text{EBM}}^{>k}$ with AUROC(\uparrow) and AUPRC(\uparrow) for CIFAR-10 (in) / SVHN (out).



Figure 7. Hierarchical sampling for Gaussian prior model (**bottom**) and EBM prior model (**top**). From **left panel to right panel**, latent vectors are sampled from the bottom layers to the top layers. Detailed sampling method can be referred in Figure 3 in [49].

We apply our joint EBM prior to BIVA [29] on CelebA-64. For training, we reuse the two-stage training scheme, where we recruit BIVA² for the first stage training and train our EBM prior model in the second stage by using the *reparametrized sampling* [44] method (similar as deep hierarchical models in Sec.5.2). We show the results of hierarchical sampling in Fig.7 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 [49], on which the improvement thus suggests that the proposed method is capable of learning hierarchical representations for multi-layer generator models. Additional results are referred to Appendix.B.1.

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 [17], we consider an unnormalized log-posterior as the decision function for EBM prior model, which is defined as

²<https://github.com/vlievin/biva-pytorch>

$$\begin{aligned} L_{\text{EBM}}^{>k} &= \mathbb{E}_{\mathbf{z}_{>k} \sim q_{\omega}(\mathbf{z}|\mathbf{x}), \mathbf{z}_{\leq k} \sim p_{\beta_{>0}, \alpha}(\mathbf{z})} [\log p_{\beta_0}(\mathbf{x}|\mathbf{z})] \\ &+ \log p_{\beta_{>0}}(\mathbf{z}) + \sum_{i=1}^L f_{\alpha_i}(\mathbf{z}_i) \end{aligned} \quad (20)$$

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 *reparametrized sampling*³ [44] with fixed inferred latent codes. With $k = 0$, all layers of latent vectors are inferred from $q_{\omega}(\mathbf{z}|\mathbf{x})$. 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 $L_{\text{EBM}}^{>0}$ and $L_{\text{EBM}}^{>k}$ as a surrogate of the likelihood-ratio which is shown to be effective for OOD detection [17]. We compute the subtraction as

$$LLR_{\text{EBM}}^{>k} = L_{\text{EBM}}^{>0} - L_{\text{EBM}}^{>k} \quad (21)$$

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.6, 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.

5.4. 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

³<https://github.com/NVlabs/VAEBM>

have oversaturated images via long-run Langevin dynamics as observed in [31]. Training an EBM that learns steady-state energy scores over realistic images can be useful but challenging.

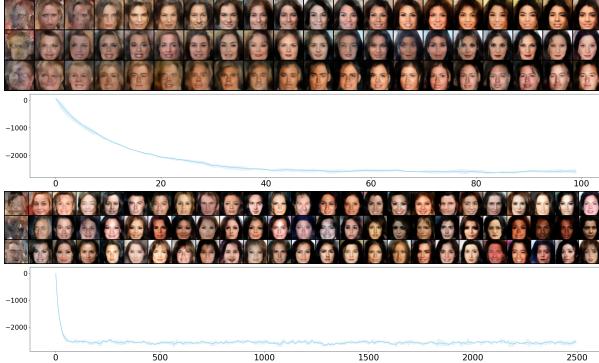


Figure 8. Trajectory in data space and energy profile in Langevin transition. **Top:** Langevin transition with 100 steps. **Bottom:** Langevin transition with 2500 steps.

We train our model on CelebA-64 using Langevin dynamic 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.8. 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.

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 $q_{\omega}(\mathbf{z}|\mathbf{x})$ could form a discriminative joint latent space that has separated probability densities for normal and anomalous data. We use un-normalized log-posterior $L_{\text{EBM}}^{>0}$ 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 [24], MEG [26], BiGAN- σ [48], OT-SRI [1], and LEBM [34] 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, 26, 34, 48].

Heldout Digit	1	4	5	7	9
VAE [24]	0.063	0.337	0.325	0.148	0.104
MEG [26]	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 [48]	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
OT-SRI [1]	0.353 \pm 0.021	0.770 \pm 0.024	0.726 \pm 0.030	0.550 \pm 0.013	0.555 \pm 0.023
LEBM [34]	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	0.470 \pm 0.009	0.941 \pm 0.001	0.964 \pm 0.003	0.815 \pm 0.004	0.796 \pm 0.004

Table 4. AUPRC scores for unsupervised anomaly detection.

5.5. Ablation Studies.

Informative prior vs. complex generator: We examine the expressivity endowed with the joint EBM prior by comparing it to hierarchical Gaussian prior model. We use the same experimental setting as reported in Tab.5 in main text and increase the complexity of generator model for hierarchical Gaussian prior. The FID results are shown in Tab.5, in which the Gaussian prior models exhibit an improvement in performance as the generator complexity increases. However, even with eight times more parameters, hierarchical Gaussian prior models still have an inferior performance compared to our joint EBM prior model.

Ours	same generator	2x parameters	4x parameters	8x parameters
28.60	42.03	39.82	37.75	36.10

Table 5. Comparison on Gaussian prior and our EBM prior.

Complexity of EBM. The energy function $f_{\alpha_i}(\mathbf{z}_i)$ is parameterized by a small multi-layer perceptron. To better understand the effectiveness of our EBM, we fix the generator network $p_{\beta_0}(\mathbf{z}|\mathbf{x})$ and increase hidden units (**nef**) of energy functions. We train our model on CIFAR-10 with **nef** increasing from 10 to 100. The results are shown in Tab.6. The larger capacity of the EBM could in general render better model performance.

nef	nef = 10	nef = 20	nef = 50	nef = 100
FID	69.73	68.45	67.88	66.32

Table 6. FID for increasing hidden units (**nef**) of EBM

MCMC sampling vs. Inference model. Two posterior sampling schemes using MCMC and inference model are compared in Tab.7 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 [14, 33], 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.

MCMC / Inf	SVHN	CelebA-64	CIFAR-10
FID	26.81 / 28.60	33.60 / 36.12	66.32 / 68.45
Time(s)	0.478 / 0.232	0.920 / 0.246	0.568 / 0.256

Table 7. FID and training time for MCMC posterior sampling and variational learning.



Figure 9. Generated images on CIFAR-10. **Left:** HVAE. FID = 79.57 **Right:** Ours. FID = 49.50

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.8. 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.

steps k	$k = 10$	$k = 20$	$k = 40$	$k = 80$	$k = 100$
FID	69.42	67.58	66.32	66.03	65.86
Time(s)	0.312	0.480	0.568	0.741	0.837

Table 8. FID and training time for increasing MCMC steps in prior sampling.

Other backbone models: We also examine the generation performance of our joint EBM prior on other multi-layer generator models, such as BIVA and HVAE. We implement the HVAE and BIVA using the provided codes⁴⁵. We show the image synthesis and corresponding FID scores in Fig.9 and Fig.10. It can be seen that the proposed method is expressive in generating sharp image synthesis and can be applied to different multi-layer generator models.

5.6. Parameter Efficiency

It is crucial to analyze the parameter complexity when comparing the generation performance. In Tab.1, we build our model with two layers of latent variables on top of the generator used in [34]. The additional layer accounts for only 1% overhead in total parameter complexity compared to LEBM [34]. For deep hierarchical models, we apply our joint EBM prior model on latent space which brings minimum overhead. The parameter complexity of the backbone NVAE and our EBM model is shown in Tab.9.

⁴<https://github.com/JakobHavtorn/hvae-oodd>

⁵<https://github.com/vlievin/biva-pytorch>



Figure 10. Generated images on CIFAR-10. **Left:** BIVA. FID = 66.37 **Right:** Ours. FID = 25.87

NVAE / EBM	CIFAR-10	CelebA-HQ-256	LSUN-Church-64
FID	39.73 / 11.34	30.25 / 9.89	38.13 / 8.38
Parameters	257M / 9M (3%)	375M / 18M (4%)	65M / 5M (7%)

Table 9. FID and parameter complexity for backbone model and EBM.

NVAE with Gaussian decoder: In addition, we also consider NVAEs with a Gaussian decoder. Note that the discrete logistic decoder aims to conditionally models the pixels of images between different channels, while Gaussian decoder is a statistical simple model that predicts pixels independently. We use the NVAE that has 30 groups on CIFAR-10 and 20 groups on CelebA-HQ-256 as used in [2, 44]. The results of FID and parameter complexity are shown in Tab.10, where our EBM prior still can largely improve the generation performance while only accounting for very small overhead in parameter complexity.

NVAE / EBM	FID	Parameters	NVAE Group
CIFAR10	52.45 / 14.92	130M / 10M (7.6%)	30
CelebA HQ 256	46.32 / 22.86	365M / 9M (2.4%)	20

Table 10. Parameter complexity and FID results based on NVAE with Gaussian decoder.

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

- [1] Dongsheng An, Jianwen Xie, and Ping Li. Learning deep latent variable models by short-run mcmc inference with optimal transport correction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15415–15424, June 2021. 8
- [2] Jyoti Aneja, Alex Schwing, Jan Kautz, and Arash Vahdat. A contrastive learning approach for training variational au-

toencoder priors. *Advances in neural information processing systems*, 34:480–493, 2021. 2, 3, 5, 6, 9

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 6

[4] Yuri Burda, Roger B. Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. 2

[5] Rewon Child. Very deep vaes generalize autoregressive models and can outperform them on images. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 1, 4

[6] Jiali Cui, Ying Nian Wu, and Tian Han. Learning hierarchical features with joint latent space energy-based prior. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2218–2227, 2023. 2

[7] Bin Dai and David Wipf. Diagnosing and enhancing vae models. *arXiv preprint arXiv:1903.05789*, 2019. 5

[8] Yilun Du, Shuang Li, Joshua Tenenbaum, and Igor Mordatch. Improved contrastive divergence training of energy based models. *arXiv preprint arXiv:2012.01316*, 2020. 2, 4, 6

[9] Yilun Du and Igor Mordatch. Implicit generation and generalization in energy-based models. *arXiv preprint arXiv:1903.08689*, 2019. 2, 4, 6

[10] Conor Durkan and Charlie Nash. Autoregressive energy machines. 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 1735–1744. PMLR, 2019. 3

[11] 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, 4

[12] Partha Ghosh, Mehdi SM Sajjadi, Antonio Vergari, Michael Black, and Bernhard Schölkopf. From variational to deterministic autoencoders. *arXiv preprint arXiv:1903.12436*, 2019. 5

[13] 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. 2

[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. 2, 5, 8

[15] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 4, 6

[16] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7978–7987, 2020. 4

[17] 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. 7, 13

[18] 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. 5

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 6

[20] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 6

[21] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in Neural Information Processing Systems*, 33:12104–12114, 2020. 6

[22] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 1

[23] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018. 6

[24] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2, 4, 5, 8, 12

[25] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[26] 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

[27] Don S Lemons and Anthony Gythiel. Paul Langevin’s 1908 paper “on the theory of brownian motion”[“sur la théorie du mouvement brownien,” cr acad. sci.(paris) 146, 530–533 (1908)]. *American Journal of Physics*, 65(11):1079–1081, 1997. 4

[28] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaou Tang. Deep learning face attributes in the wild. *CoRR*, abs/1411.7766, 2014. 5

[29] 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, 4, 7, 13

[30] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural

images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. 5

[31] 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. 8

[32] 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. 2

[33] 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, 2, 5, 8

[34] 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, 5, 8, 9, 14

[35] Masaki Saito, Shunta Saito, Masanori Koyama, and Sosuke Kobayashi. Train sparsely, generate densely: Memory-efficient unsupervised training of high-resolution temporal gan. *International Journal of Computer Vision*, 128(10):2586–2606, 2020. 1

[36] 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. 1, 2, 4, 5

[37] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019. 6

[38] 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

[39] Jakub Tomczak and Max Welling. Vae with a vampprior. In *International Conference on Artificial Intelligence and Statistics*, pages 1214–1223. PMLR, 2018. 5

[40] 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

[41] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *Advances in Neural Information Processing Systems*, 33:19667–19679, 2020. 1, 4, 6, 13

[42] Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*, pages 1747–1756. PMLR, 2016. 6

[43] Zhisheng Xiao and Tian Han. Adaptive multi-stage density ratio estimation for learning latent space energy-based model. *arXiv preprint arXiv:2209.08739*, 2022. 5

[44] 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, 4, 5, 6, 7, 9, 13

[45] Jianwen Xie, Yaxuan Zhu, Jun Li, and Ping Li. A tale of two flows: Cooperative learning of langevin flow and normalizing flow toward energy-based model. In *International Conference on Learning Representations*, 2022. 4

[46] Xuwang Yin, Shiying Li, and Gustavo K Rohde. Analyzing and improving generative adversarial training for generative modeling and out-of-distribution detection. *arXiv preprint arXiv:2012.06568*, 2020. 2, 4, 6

[47] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 6

[48] 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

[49] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Learning hierarchical features from generative models. *arXiv preprint arXiv:1702.08396*, 2017. 1, 7

A. Theoretical Derivations

A.1. Maximum Likelihood Estimation

Recall that $\nabla_\theta \log p_\theta(\mathbf{x}) = \mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{x})}[\nabla_\theta \log p_{\beta_0}(\mathbf{x}|\mathbf{z})] + \mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{x})}[\nabla_\theta \log p_{\alpha, \beta_{>0}}(\mathbf{z})]$, where $\theta = (\alpha, \beta_0, \beta_{>0})$. For the learning gradient of prior model $(\alpha_i, \beta_{>0})$, we compute $\mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{x})}[\nabla_{\alpha_i, \beta_{>0}} \log p_{\alpha, \beta_{>0}}(\mathbf{z})]$ as

$$\begin{aligned} \nabla_{\alpha_i} \log p_\theta(\mathbf{x}) &= \mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{x})}[\nabla_{\alpha_i} \log p_{\alpha, \beta_{>0}}(\mathbf{z})] \quad (22) \\ &= \mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{x})}[\nabla_{\alpha_i} f_{\alpha_i}(\mathbf{z}_i)] - \nabla_{\alpha_i} \log Z_{\alpha, \beta_{>0}} \end{aligned}$$

$$\begin{aligned} \nabla_{\beta_i} \log p_\theta(\mathbf{x}) &= \mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{x})}[\nabla_{\beta_i} \log p_{\alpha, \beta_{>0}}(\mathbf{z})] \quad (23) \\ &= \mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{x})}[\nabla_{\beta_i} \log p_{\beta_i}(\mathbf{z}_i|\mathbf{z}_{i+1})] - \nabla_{\beta_i} \log Z_{\alpha, \beta_{>0}} \end{aligned}$$

where $Z_{\alpha, \beta_{>0}} = \int \exp[f_\alpha(\mathbf{z})] p_{\beta_{>0}}(\mathbf{z}) d\mathbf{z}$. Therefore, for $\nabla_{\alpha_i} \log Z_{\alpha, \beta_{>0}}$, we have

$$\begin{aligned} &\nabla_{\alpha_i} \log Z_{\alpha, \beta_{>0}} \quad (24) \\ &= \frac{1}{Z_{\alpha, \beta_{>0}}} \int \nabla_{\alpha_i} \exp \left[\sum_{i=1}^L f_{\alpha_i}(\mathbf{z}_i) \right] p_{\beta_{>0}}(\mathbf{z}) d\mathbf{z} \\ &= \int p_{\alpha, \beta_{>0}}(\mathbf{z}) \nabla_{\alpha_i} f_{\alpha_i}(\mathbf{z}_i) d\mathbf{z} \\ &= \mathbb{E}_{p_{\alpha, \beta_{>0}}(\mathbf{z})}[\nabla_{\alpha_i} f_{\alpha_i}(\mathbf{z}_i)] \end{aligned}$$

For $\nabla_{\beta_{>0}} \log Z_{\alpha, \beta_{>0}}$, we have

$$\begin{aligned} &\nabla_{\beta_i} \log Z_{\alpha, \beta_{>0}} \quad (25) \\ &= \frac{1}{Z_{\alpha, \beta_{>0}}} \int \exp[f_\alpha(\mathbf{z})] \nabla_{\beta_i} \prod_{i=1}^{L-1} p_{\beta_i}(\mathbf{z}_i|\mathbf{z}_{i+1}) p(\mathbf{z}_L) d\mathbf{z} \\ &= \int p_{\alpha, \beta_{>0}}(\mathbf{z}) \nabla_{\beta_i} \log p_{\beta_i}(\mathbf{z}_i|\mathbf{z}_{i+1}) d\mathbf{z} \\ &= \mathbb{E}_{p_{\alpha, \beta_{>0}}(\mathbf{z})}[\nabla_{\beta_i} \log p_{\beta_i}(\mathbf{z}_i|\mathbf{z}_{i+1})] \end{aligned}$$

By applying Eqn.24 to Eqn.22, we have

$$\begin{aligned} \nabla_{\alpha_i} \log p_\theta(\mathbf{x}) &= \mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{x})}[\nabla_{\alpha_i} f_{\alpha_i}(\mathbf{z}_i)] \quad (26) \\ &\quad - \mathbb{E}_{p_{\alpha, \beta_{>0}}(\mathbf{z})}[\nabla_{\alpha_i} f_{\alpha_i}(\mathbf{z}_i)] \end{aligned}$$

By applying Eqn.25 and Eqn.23, we have

$$\begin{aligned} \nabla_{\beta_i} \log p_\theta(\mathbf{x}) &= \mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{x})}[\nabla_{\beta_i} \log p_{\beta_i}(\mathbf{z}_i|\mathbf{z}_{i+1})] \quad (27) \\ &\quad - \mathbb{E}_{p_{\alpha, \beta_{>0}}(\mathbf{z})}[\nabla_{\beta_i} \log p_{\beta_i}(\mathbf{z}_i|\mathbf{z}_{i+1})] \end{aligned}$$

A.2. Variational Learning

Recall that $L(\theta, \omega) = D_{\text{KL}}(q_\omega(\mathbf{x}, \mathbf{z}) || p_\theta(\mathbf{x}, \mathbf{z}))$. We can view such joint KL as a surrogate of the MLE objective with the KL perturbation term, i.e., $L(\theta, \omega) = D_{\text{KL}}(p_{\text{data}}(\mathbf{x}) || p_\theta(\mathbf{x})) + D_{\text{KL}}(q_\omega(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z}|\mathbf{x}))$. Specifically, we have

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

where $C \equiv -H(p_{\text{data}}(\mathbf{x}))$ is the entropy of the empirical data distribution and can be treated as constant. $\tilde{L}(\theta, \omega)$ is a lower bound of the log-likelihood $\log p_\theta(\mathbf{x})$ typically known as ELBO [24]. Notice that, with the joint EBM prior model, we consider the KL optimization between the aggregate posterior and EBM prior model, i.e., $\tilde{L}(\theta, \omega) = \mathbb{E}_{q_\omega(\mathbf{z}|\mathbf{x})}[\log p_{\beta_0}(\mathbf{x}|\mathbf{z})] - D_{\text{KL}}(q_\omega(\mathbf{z}|\mathbf{x}) || p_{\alpha, \beta_{>0}}(\mathbf{z}))$, while VAEs compute $D_{\text{KL}}(q_\omega(\mathbf{z}|\mathbf{x}) || p_{\beta_{>0}}(\mathbf{z}))$, where $p_{\beta_{>0}}(\mathbf{z})$ is the Gaussian prior model.

Therefore, we can compute the gradient $\nabla_{\theta, \omega} \tilde{L}(\theta, \omega)$ to jointly update the inference, generator and EBM prior model. Learning the prior model $(\alpha_i, \beta_{>0})$ involves computing the derivative of $\log Z_{\alpha, \beta_{>0}}$, which can be referred to Eqn.24 and Eqn.25.

A.3. Change of Variable

We observe that using Langevin dynamic on latent space for deep hierarchical structures can be heterogeneous, where latent variables may be formed in different shapes (e.g., spatial variables and vectors) and can rely on the distribution that has a high variance. Therefore, we further consider $\epsilon_{\mathbf{z}}$ -space, which has a unit variance and can make the prior sampling more efficient and effective. For brevity, we take a two-layer structure as an example, i.e., $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2)$, where for L layers, the derivation is the same.

Deterministic transformation $T_{\beta_{>0}}$: For generator model $p_{\beta_{>0}}(\mathbf{z}_1, \mathbf{z}_2)$, \mathbf{z}_1 follows conditional Gaussian distribution as $p(\mathbf{z}_1|\mathbf{z}_2) \sim \mathcal{N}(\mu_{\beta_1}(\mathbf{z}_2), \sigma_{\beta_1}(\mathbf{z}_2))$, while $p(\mathbf{z}_2)$ is assumed to be unit Gaussian, such that $p(\mathbf{z}_2) \sim \mathcal{N}(0, I_d)$. Let $(\epsilon_{\mathbf{z}_1}, \epsilon_{\mathbf{z}_2})$ be the re-parametrization variables, we have $T_{\beta_{>0}}$ defined as

$$\mathbf{z}_2 = T_{\beta_{>0}}^{\mathbf{z}_2}(\epsilon_{\mathbf{z}_2}) = \epsilon_{\mathbf{z}_2} \quad (28)$$

$$\mathbf{z}_1 = T_{\beta_{>0}}^{\mathbf{z}_1}(\epsilon_{\mathbf{z}_1}, \epsilon_{\mathbf{z}_2}) = \mu_{\beta_1}(\mathbf{z}_2) + \sigma_{\beta_1}(\mathbf{z}_2) \cdot \epsilon_{\mathbf{z}_1} \quad (29)$$

$T_{\beta_{>0}}^{\mathbf{z}_2}(\epsilon_{\mathbf{z}_2})$ and $T_{\beta_{>0}}^{\mathbf{z}_1}(\epsilon_{\mathbf{z}_1}, \epsilon_{\mathbf{z}_2})$ are invertible and usually referred as reparameterization trick used in VAEs. Thus, the re-parametrization variables $(\epsilon_{\mathbf{z}_1}, \epsilon_{\mathbf{z}_2})$ can be independently drawn from Gaussian noise, i.e., $(\epsilon_{\mathbf{z}_1}, \epsilon_{\mathbf{z}_2}) \sim p_\epsilon(\epsilon_{\mathbf{z}_1}, \epsilon_{\mathbf{z}_2})$, where $p_\epsilon(\epsilon_{\mathbf{z}_1}, \epsilon_{\mathbf{z}_2}) = p_{\epsilon_1}(\epsilon_{\mathbf{z}_1})p_{\epsilon_2}(\epsilon_{\mathbf{z}_2})$ and $p_{\epsilon_i}(\epsilon_{\mathbf{z}_i}) \sim \mathcal{N}(0, I_{\|\epsilon_{\mathbf{z}_i}\|})$.

Toward $\epsilon_{\mathbf{z}}$ -space $p_{\alpha, \beta_{>0}}(\epsilon_{\mathbf{z}_1}, \epsilon_{\mathbf{z}_2})$: With invertible transformation $T_{\beta_{>0}}$, we can apply change of variable rule as

$$p_{\beta_{>0}}(\mathbf{z}_1, \mathbf{z}_2) = p_{\epsilon}(\epsilon_{\mathbf{z}_1}, \epsilon_{\mathbf{z}_2}) |\det(J_{T_{\beta_{>0}}^{-1}})| \quad (30)$$

$$p_{\epsilon}(\epsilon_{\mathbf{z}_1}, \epsilon_{\mathbf{z}_2}) = p_{\beta_{>0}}(\mathbf{z}_1, \mathbf{z}_2) |\det(J_{T_{\beta_{>0}}})| \quad (31)$$

where $J_{T_{\beta_{>0}}}$ is the Jacobian of $T_{\beta_{>0}}$.

For brevity, we denote $\epsilon_{\mathbf{z}} = (\epsilon_{\mathbf{z}_1}, \epsilon_{\mathbf{z}_2})$, then $p_{\beta_{>0}}(\mathbf{z}) = p_{\epsilon}(\epsilon_{\mathbf{z}}) |\det(J_{T_{\beta_{>0}}^{-1}})|$ and $p_{\epsilon}(\epsilon_{\mathbf{z}}) = p_{\beta_{>0}}(\mathbf{z}) |\det(J_{T_{\beta_{>0}}})|$. Recall that the proposed joint EBM prior model is defined as $p_{\alpha, \beta_{>0}}(\mathbf{z})$. With change of variable, $p_{\alpha, \beta_{>0}}(\epsilon_{\mathbf{z}})$ is

$$\begin{aligned} p_{\alpha, \beta_{>0}}(\epsilon_{\mathbf{z}}) &= p_{\alpha, \beta_{>0}}(\mathbf{z}) |\det(J_{T_{\beta_{>0}}})| \\ &= \frac{1}{Z_{\alpha, \beta_{>0}}} \exp f_{\alpha}(T_{\beta_{>0}}(\epsilon_{\mathbf{z}})) p_{\beta_{>0}}(\mathbf{z}) |\det(J_{T_{\beta_{>0}}})| \\ &= \frac{1}{Z_{\alpha, \beta_{>0}}} \exp f_{\alpha}(T_{\beta_{>0}}(\epsilon_{\mathbf{z}})) p_{\epsilon}(\epsilon_{\mathbf{z}}) \end{aligned}$$

Therefore, sampling from $p_{\alpha, \beta_{>0}}(\mathbf{z})$ can be done by first sampling $\epsilon_{\mathbf{z}}$ from $p_{\alpha, \beta_{>0}}(\epsilon_{\mathbf{z}})$ and then using deterministic transformation $T_{\beta_{>0}}$ to obtain \mathbf{z} as Eqn.28 and Eqn.29. Compared to latent space $p_{\alpha, \beta_{>0}}(\mathbf{z})$, the $\epsilon_{\mathbf{z}}$ -space $p_{\alpha, \beta_{>0}}(\epsilon_{\mathbf{z}})$ independently draws samples from the same Gaussian distribution, and such distribution has a unit variance allowing us to use the fixed step size of Langevin dynamic to efficiently and effectively explore the latent space at different layers for deep hierarchical structures. For experiments with backbone model BIVA [29] or NVAE [41], we adopt similar reparametrized sampling scheme as VAEBM [44] via public code⁶.

B. Additional Experiments

B.1. Hierarchical Representations

Hierarchical reconstruction. To examine the hierarchical representation, we further conduct hierarchical reconstruction by replacing the inferred latent vectors at the bottom layers with the ones from the prior distribution. We use BIVA [29] as our backbone model for multi-layer generator and inference model, and we use Langevin dynamic for prior sampling. Specifically, we run prior Langevin sampling for the latent codes at lower layers (e.g., $\mathbf{z}_{i \leq k}$) with the latent codes at top layers (from BIVA inference model) remaining fixed (using Eqn.20 in main text). We train our model on CelebA-64 and show hierarchical reconstructions in Fig.12.

We observe that the details in reconstructions can be gradually replaced by common features as more layers of latent variables are sampled from the prior distribution. For example, the sunglass first becomes a more common glass



Figure 11. Hierarchical sampling with NVAE backbone on CelebA-HQ-256.

and then eventually disappears. This concurs with the observation in [17], suggesting that our model carries different levels of abstract representations within the hierarchical structure.



(a) Example. (b) Sampling from bottom layer to top layer.

Figure 12. Hierarchical reconstruction

Additional results for OOD detection: In addition, we compute AUROC, AUPRC and FPR80 for BIVA and our EBM prior model in OOD detection. We use the log-likelihood $L^{>k}$ and a ratio type $LLR^{>k}$ [17] as the decision functions for BIVA. If the low-level representations are well-learned at the bottom layers, using decision function with higher k should render better detection performance for reducing impact of shared low-level features. The results are shown in Tab.11.

BIVA / Ours	AUROC↑	AUPRC↑	FPR80↓
$L^{>0} / L_{\text{EBM}}^{>0}$	0.066 / 0.087	0.339 / 0.319	0.997 / 0.999
$L^{>3} / L_{\text{EBM}}^{>3}$	0.307 / 0.324	0.427 / 0.438	0.970 / 0.972
$L^{>6} / L_{\text{EBM}}^{>6}$	0.436 / 0.449	0.514 / 0.528	0.942 / 0.942
$L^{>9} / L_{\text{EBM}}^{>9}$	0.866 / 0.870	0.855 / 0.858	0.230 / 0.227
$LLR^{>9} / LLR_{\text{EBM}}^{>9}$	0.885 / 0.927	0.876 / 0.918	0.200 / 0.113

Table 11. AUROC, AUPRC and FPR80 for BIVA and our EBM prior model on CIFAR10(in) / SVHN(out).

⁶<https://github.com/NVlabs/VAEBM>

C. Experiment Details

Fréchet Inception Distance: We compute FID scores with 30,000 generated images for CelebA-HQ-256 and 50,000 generated images for other data.

Implementations: For comparisons in generator models with informative prior, we train our model on SVHN (32 x 32), CIFAR-10 (32 x 32), and CelebA-64 (64 x 64), where we use full training split of SVHN and CIFAR-10 and 40,000 cropped training examples of CelebA-64 following the protocol in [34]. All training images are resized and scaled to [-1, 1]. For applying to NVAE backbone models, we train our joint EBM prior on latent variables of all layers. The implementations of models on CelebA-64 and EBMs for NVAE backbone are shown in Tab.12. We denote the operation of convolution and transposed convolution as $\text{conv}(k, c, s)$ and $\text{convT}(k, c, s)$, where k is the kernel size, c is the channel number and s is the stride number, and we denote LeakyReLU as LReLU.

D. Additional qualitative results:

We show additional image synthesis for CIFAR-10, LSUN-Church-64 and CelebA-HQ-256 in Fig.13, Fig.15, Fig.17 and Fig.18. The additional visualizations of langevin transition that starts from $p_{\beta>0}(\mathbf{z})$ toward the learned EBM prior distribution $p_{\alpha,\beta>0}(\mathbf{z})$ are shown in Fig.14, Fig.16 and Fig.19.

Layers		In-Out Size
EBM $f_{\alpha_i}(\mathbf{z}_i)$ for NVAE backbone		
Input: \mathbf{z}_i		(h x w x c)
$N \times \text{conv}(4, 64, 2)$, LReLU		(4 x 4 x 64)
$N \times \text{Linear}(200)$, LReLU		200
Linear(1)		1
Generator Model $p_{\beta_1}(\mathbf{z}_1 \mathbf{z}_2)$		
Input: \mathbf{z}_2		100
$\text{Linear}(200)$, LReLU		200
$\text{Linear}(200)$, LReLU		200
$\text{Linear}(200)$		200
Split for $\mu_{\mathbf{z}_1}$ and $\log \sigma_{\mathbf{z}_1}$		100, 100
Generator Model $p_{\beta_0}(\mathbf{x} \mathbf{z})$		
Input: \mathbf{z}_1		(1 x 1 x 100)
$\text{convT}(4, 1024, 1)$, LReLU		(4 x 4 x 1024)
$\text{convT}(4, 512, 2)$, LReLU		(8 x 8 x 512)
$\text{convT}(4, 256, 2)$, LReLU		(16 x 16 x 256)
$\text{convT}(4, 128, 2)$, LReLU		(32 x 32 x 128)
$\text{convT}(4, 3, 2)$, Tanh		(64 x 64 x 3)
Inference Model $q_{\omega_2}(\mathbf{z}_2 \mathbf{z}_1)$		
Input: \mathbf{z}_1		100
$\text{Linear}(200)$, LReLU		200
$\text{Linear}(200)$, LReLU		200
$\text{Linear}(200)$		200
Split for $\mu_{\mathbf{z}_2}$ and $\log \sigma_{\mathbf{z}_2}$		100, 100
Inference Model $q_{\omega_1}(\mathbf{z}_1 \mathbf{x})$		
Input: \mathbf{x}		(64 x 64 x 3)
$\text{conv}(4, 128, 2)$, LReLU		(32 x 32 x 128)
$\text{conv}(4, 256, 2)$, LReLU		(16 x 16 x 256)
$\text{conv}(4, 512, 2)$, LReLU		(8 x 8 x 512)
$\text{conv}(4, 1024, 2)$, LReLU		(4 x 4 x 1024)
$\text{conv}(4, 200, 1)$		(1 x 1 x 200)
Split for $\mu_{\mathbf{z}_1}$ and $\log \sigma_{\mathbf{z}_1}$		100, 100
EBM $f_{\alpha_1}(\mathbf{z}_1)$		
Input: \mathbf{z}_1		100
$\text{Linear}(200)$, LReLU		200
Linear(1)		1
EBM $f_{\alpha_2}(\mathbf{z}_2)$		
Input: \mathbf{z}_2		100
$\text{Linear}(100)$, LReLU		100
$\text{Linear}(100)$, LReLU		100
Linear(1)		1

Table 12. Network structures for generation, inference and EBMs on CELEBA-64 and EBM structure for NVAE backbone models.

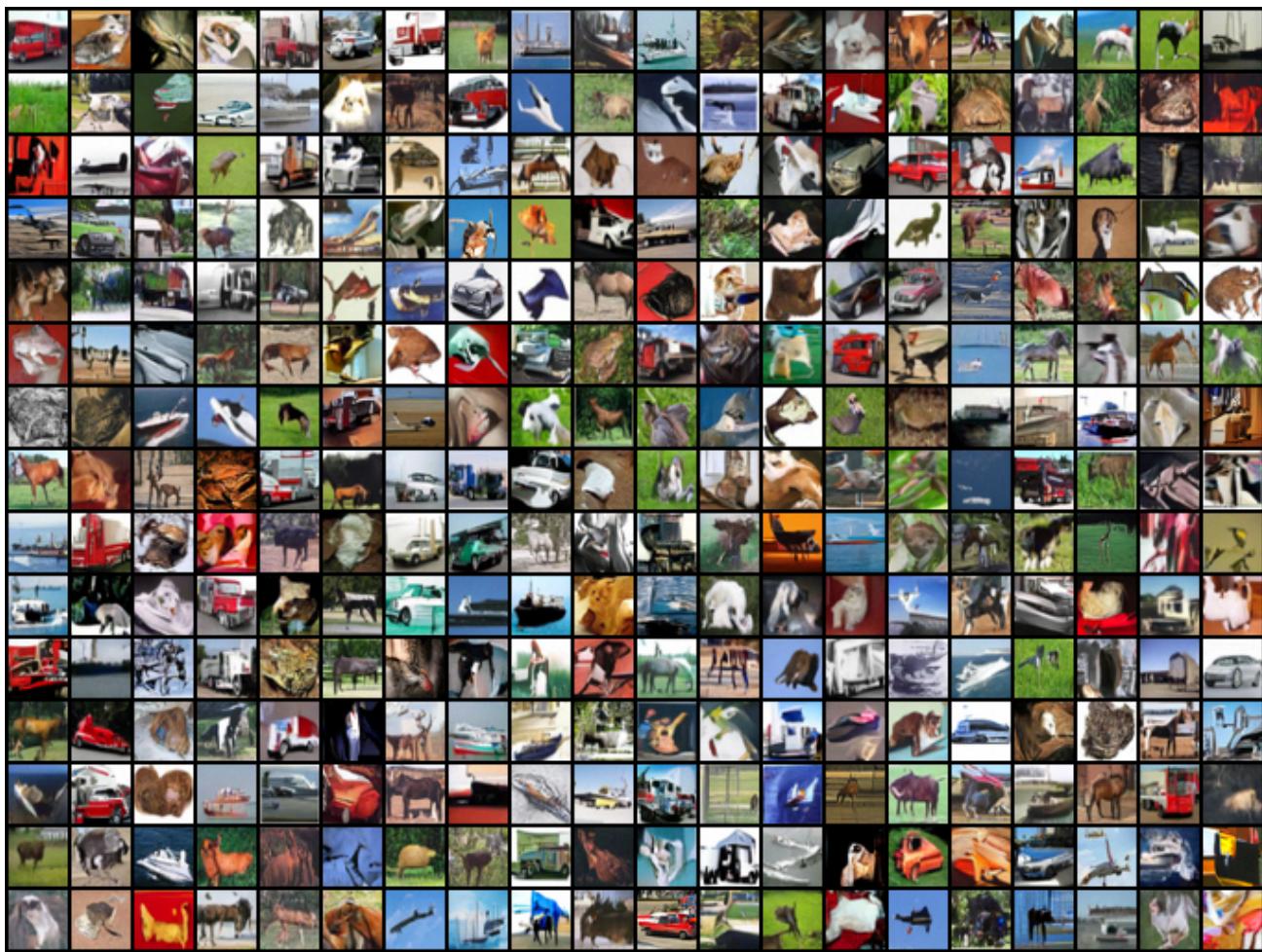


Figure 13. Generated images on CIFAR-10. Samples are uncurated.

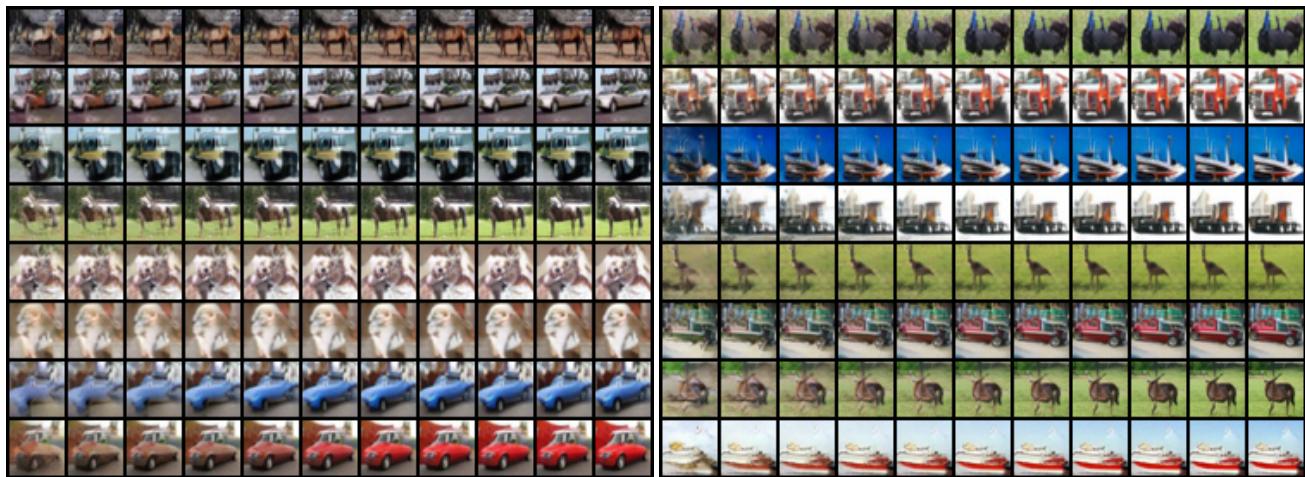


Figure 14. Langevin transition on CIFAR-10.



Figure 15. Generated images on LSUN-Church-64. Samples are uncurated.

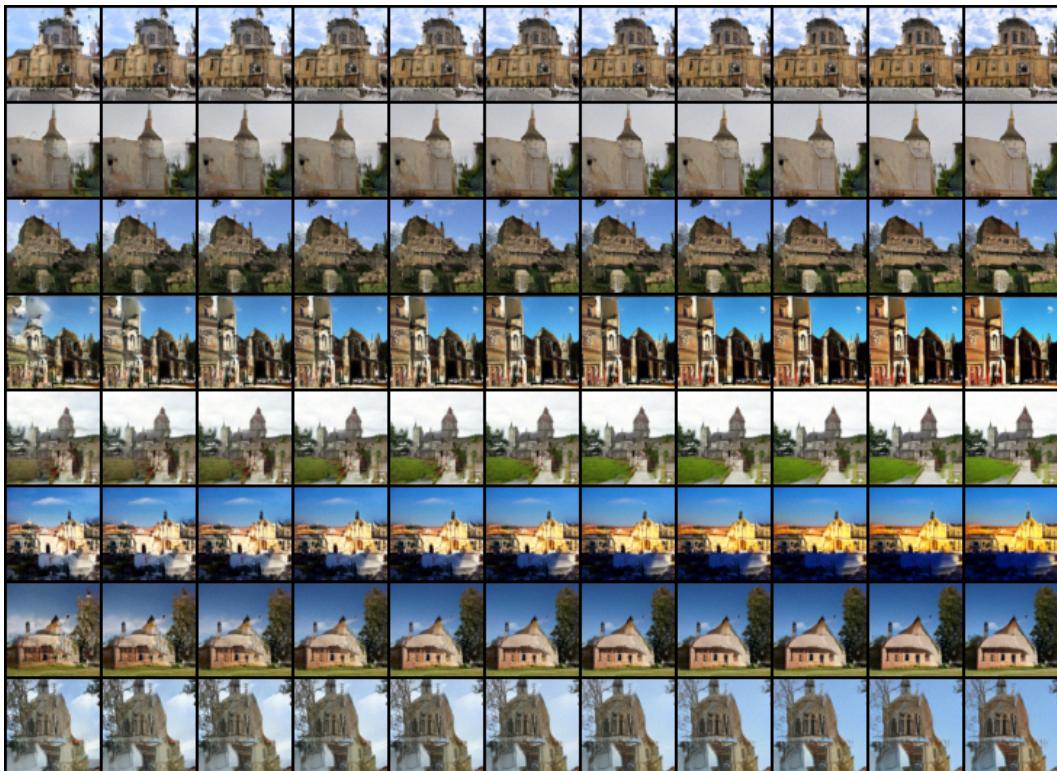


Figure 16. Langevin transition on LSUN-Church-64.

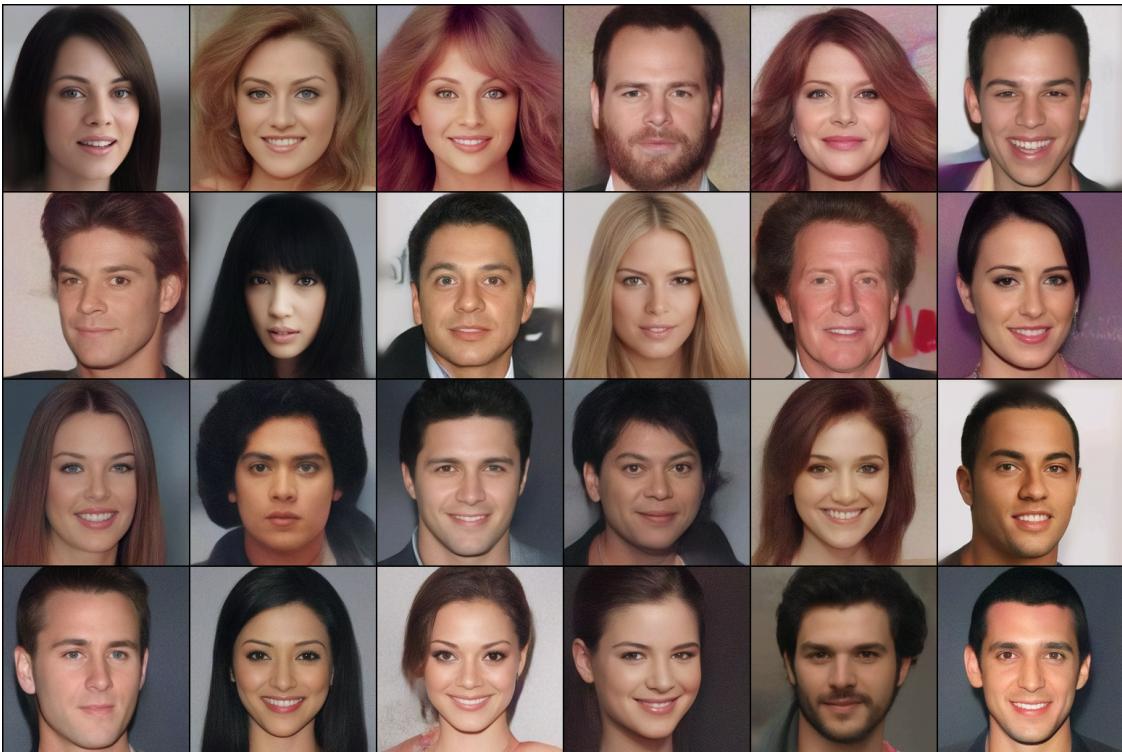


Figure 17. Generated images on CelebA-HQ-256 (temperature $t=0.7$). Samples are uncurated.

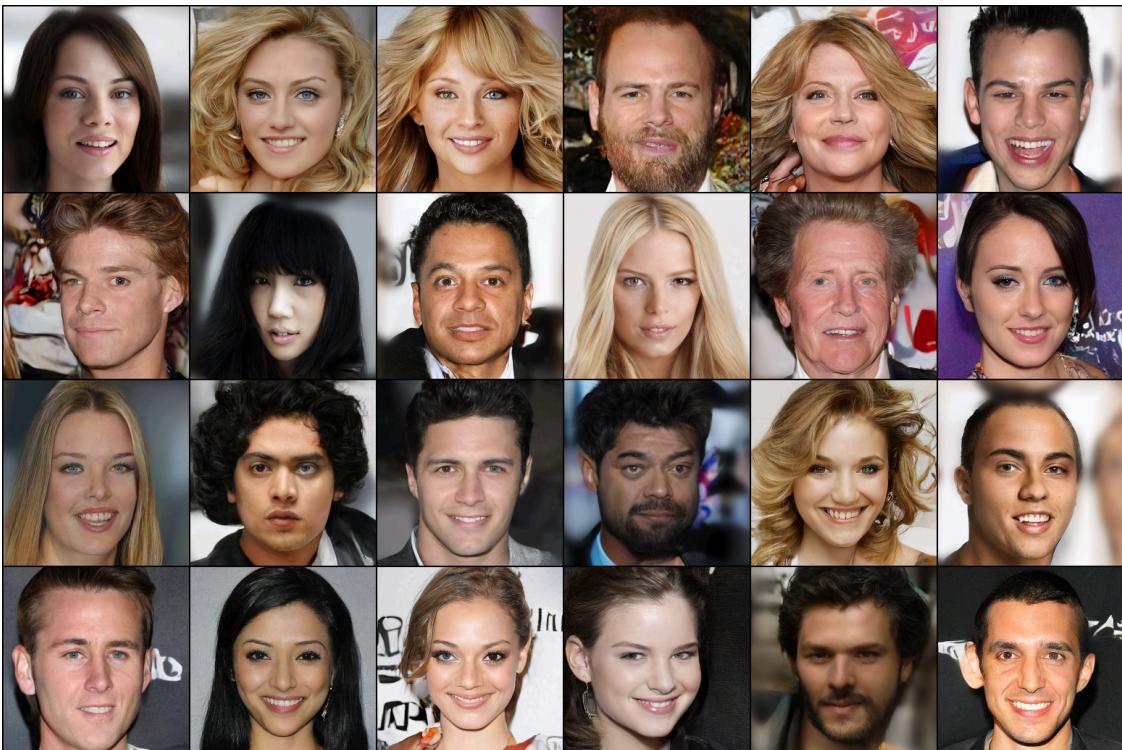


Figure 18. Generated images on CelebA-HQ-256 (temperature $t=1.0$). Samples are uncurated.



Figure 19. Langevin transition on CelebA-HQ-256. (temperature $t=1.0$).