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

Jiali Cui[1], Ying Nian Wu[2], Tian Han[1]
[1]Department of Computer Science, Stevens Institute of Technology
[2]Department of Statistics, University of California, Los Angeles
{jcui7,than6}@stevens.edu, ywu@stat.ucla.edu

1. Theoretical Derivations

1.1. 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

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

$$\begin{aligned} &\nabla_{\beta_i} \log p_\theta(\mathbf{x}) = \mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{x})}[\nabla_{\beta_i} \log p_{\alpha,\beta_{>0}}(\mathbf{z})] \quad (2)\\ =\ & \mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{x})}[\nabla_{\beta_i} \log p_{\beta_i}(\mathbf{z}_i|\mathbf{z}_{i+1})] - \nabla_{\beta_i} \log \mathrm{Z}_{\alpha,\beta_{>0}} \end{aligned}$$

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

$$\begin{aligned} &\nabla_{\alpha_i} \log \mathrm{Z}_{\alpha,\beta_{>0}} \quad (3)\\ =\ & \frac{1}{\mathrm{Z}_{\alpha,\beta_{>0}}} \int \nabla_{\alpha_i} \exp[\sum_{i=1}^{L} f_{\alpha_i}(\mathbf{z}_i)] p_{\beta_{>0}}(\mathbf{z}) d\mathbf{z}\\ =\ & \int p_{\alpha,\beta_{>0}}(\mathbf{z}) \nabla_{\alpha_i} f_{\alpha_i}(\mathbf{z}_i) d\mathbf{z}\\ =\ & \mathbb{E}_{p_{\alpha,\beta_{>0}}(\mathbf{z})}[\nabla_{\alpha_i} f_{\alpha_i}(\mathbf{z}_i)] \end{aligned}$$

For $\nabla_{\beta_{>0}} \log \mathrm{Z}_{\alpha,\beta_{>0}}$, we have

$$\begin{aligned} &\nabla_{\beta_i} \log \mathrm{Z}_{\alpha,\beta_{>0}} \quad (4)\\ =\ & \frac{1}{\mathrm{Z}_{\alpha,\beta_{>0}}} \int \exp[f_\alpha(\mathbf{z})] \nabla_{\beta_i} \prod_{i=1}^{L-1} p_{\beta_i}(\mathbf{z}_i|\mathbf{z}_{i+1}) p(\mathbf{z}_L) d\mathbf{z}\\ =\ & \int p_{\alpha,\beta_{>0}}(\mathbf{z}) \nabla_{\beta_i} \log p_{\beta_i}(\mathbf{z}_i|\mathbf{z}_{i+1}) d\mathbf{z}\\ =\ & \mathbb{E}_{p_{\alpha,\beta_{>0}}(\mathbf{z})}[\nabla_{\beta_i} \log p_{\beta_i}(\mathbf{z}_i|\mathbf{z}_{i+1})] \end{aligned}$$

By applying Eqn.3 to Eqn.1, we have

$$\begin{aligned} \nabla_{\alpha_i} \log p_\theta(\mathbf{x}) &= \mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{x})}[\nabla_{\alpha_i} f_{\alpha_i}(\mathbf{z}_i)] \quad (5)\\ &- \mathbb{E}_{p_{\alpha,\beta_{>0}}(\mathbf{z})}[\nabla_{\alpha_i} f_{\alpha_i}(\mathbf{z}_i)] \end{aligned}$$

By applying Eqn.4 and Eqn.2, we have

$$\begin{aligned} \nabla_{\beta_i} \log p_\theta(\mathbf{x}) &= \mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{x})}[\nabla_{\beta_i} \log p_{\beta_i}(\mathbf{z}_i|\mathbf{z}_{i+1})] \quad (6)\\ &- \mathbb{E}_{p_{\alpha,\beta_{>0}}(\mathbf{z})}[\nabla_{\beta_i} \log p_{\beta_i}(\mathbf{z}_i|\mathbf{z}_{i+1})] \end{aligned}$$

1.2. Variational Learning

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

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

where $C \equiv -H(p_{\mathrm{data}}(x))$ 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 [3]. Notice that, with the joint EBM prior model, we consider the KL optimization between the aggregate posterior and EBM prior model, i.e., $\tilde{L}(\theta, \omega) = \mathbb{E}_{q_\omega(\mathbf{z}|\mathbf{x})}[\log p_{\beta_0}(\mathbf{x}|\mathbf{z})] - D_{\mathrm{KL}}(q_\omega(\mathbf{z}|\mathbf{x})||p_{\alpha,\beta_{>0}}(\mathbf{z}))$, while VAEs compute $D_{\mathrm{KL}}(q_\omega(\mathbf{z}|\mathbf{x})||p_{\beta_{>0}}(\mathbf{z}))$, 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 $\log \mathrm{Z}_{\alpha,\beta_{>0}}$, which can be referred to Eqn.3 and Eqn.4.

1.3. 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

$$\mathbf{z}_2 = T^{\mathbf{z}_2}_{\beta_{>0}}(\epsilon_{\mathbf{z}_2}) = \epsilon_{\mathbf{z}_2} \quad (7)$$

$$\mathbf{z}_1 = T^{\mathbf{z}_1}_{\beta_{>0}}(\epsilon_{\mathbf{z}_1}, \epsilon_{\mathbf{z}_2}) = \mu_{\beta_1}(\mathbf{z}_2) + \sigma_{\beta_1}(\mathbf{z}_2) \cdot \epsilon_{\mathbf{z}_1} \quad (8)$$

$T^{\mathbf{z}_2}_{\beta_{>0}}(\epsilon_{\mathbf{z}_2})$ and $T^{\mathbf{z}_1}_{\beta_{>0}}(\epsilon_{\mathbf{z}_1}, \epsilon_{\mathbf{z}_2})$ 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

$$p_{\beta_{>0}}(\mathbf{z}_1, \mathbf{z}_2) = p_\epsilon(\epsilon_{\mathbf{z}_1}, \epsilon_{\mathbf{z}_2}) |\det(J_{T^{-1}_{\beta_{>0}}})| \quad (9)$$

$$p_\epsilon(\epsilon_{\mathbf{z}_1}, \epsilon_{\mathbf{z}_2}) = p_{\beta_{>0}}(\mathbf{z}_1, \mathbf{z}_2) |\det(J_{T_{\beta_{>0}}})| \quad (10)$$

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 $p_{\beta_{>0}}(\mathbf{z}) = p_\epsilon(\epsilon_{\mathbf{z}}) |\det(J_{T^{-1}_{\beta_{>0}}})|$ 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

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

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.7 and Eqn.8. 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 [4] or NVAE [6], we adopt similar reparametrized sampling scheme as VAEBM [7] via public code[1].

[1] https://github.com/NVlabs/VAEBM

2. Additional Experiments

2.1. Analysis of EBM prior

Latent visualization: To better understand the effectiveness of the proposed EBM prior model, 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. We visualize the transition of Langevin dynamics on each layer in Fig.1, where latent variables can be successfully tilted via EBM to match the multi-modal posterior, which suggests the expressiveness of our EBM prior.

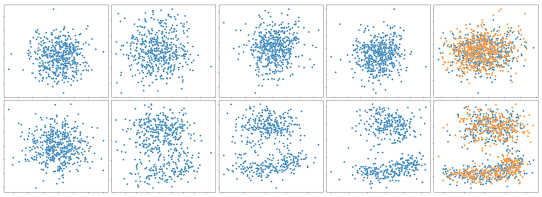


Figure 1. Langevin transition on latent codes (bottom: $\mathbf{z}_1$, top: $\mathbf{z}_2$). **Blue**, **Orange** color indicate prior and posterior, respectively.

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

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.2, 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 2. Comparison on Gaussian prior and our EBM prior.

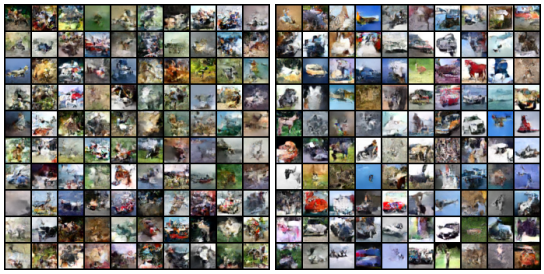


Figure 2. Generated images on CIFAR-10. **Left:** HVAE. FID = 79.57 **Right:** Ours. FID = 49.50

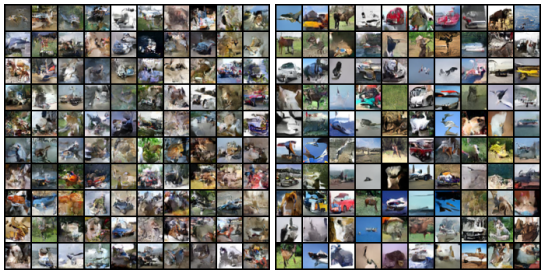


Figure 3. Generated images on CIFAR-10. **Left:** BIVA. FID = 66.37 **Right:** Ours. FID = 25.87

2.2. Image Synthesis

NVAE with Gaussian decoder: We use the NVAE[2] that has a mixture discrete logistic decoder in the main text for CIFAR-10 and CelebA-HQ-256. 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 [1, 7]. The results of FID and parameter complexity are shown in Tab.3, 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 3. Parameter complexity and FID results based on NVAE with Gaussian decoder.

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 codes[3][4]. We show the image synthesis and corresponding FID scores in Fig.2 and Fig.3. 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.

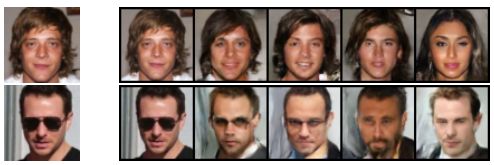
2.3. Hierarchical Representations

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 [4] 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.5.

[2]https://github.com/NVlabs/NVAE

[3]https://github.com/JakobHavtorn/hvae-oodd

[4]https://github.com/vlievin/biva-pytorch



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

Figure 5. Hierarchical reconstruction

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 and then eventually disappears. This concurs with the observation in [2], suggesting that our model carries different levels of abstract representations within the hierarchical structure.

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 $LLR^{>k}$ [2] 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.4.

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

Table 4. AUROC, AUPRC and FPR80 for BIVA and our EBM prior model on CIFAR10(in) / SVHN(out).

3. 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 [5]. 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.5. We denote the operation of convolution and transposed convolution as $\mathrm{conv(k, c, s)}$ and $\mathrm{convT(k, c, s)}$, where k is the kernel size, c is the channel number and s is the stride number, and we denote $\mathrm{LeakyReLU}$ as LReLU.

4. Additional qualitative results:

We show additional image synthesis for CIFAR-10, LSUN-Church-64 and CelebA-HQ-256 in Fig.6, Fig.8, Fig.10 and Fig.11. 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.7 and Fig.9.

Layers	In-Out Size
EBM $f_{\alpha_i}(\mathbf{z}_i)$ for NVAE backbone	
Input: $\mathbf{z}_i$	(h x w x c)
N x conv (4, 64, 2), LReLU	(4 x 4 x 64)
N x Linear (200), LReLU	200
Linear (1)	1
Generator Model $p_{\beta_1}(\mathbf{z}_1\|\mathbf{z}_2)$	
Input: $\mathbf{z}_2$	100
Linear (200), LReLU	200
Linear (200), LReLU	200
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)
convT (4, 1024, 1), LReLU	(4 x 4 x 1024)
convT (4, 512, 2), LReLU	(8 x 8 x 512)
convT (4, 256, 2), LReLU	(16 x 16 x 256)
convT (4, 128, 2), LReLU	(32 x 32 x 128)
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
Linear (200), LReLU	200
Linear (200), LReLU	200
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)
conv (4, 128, 2), LReLU	(32 x 32 x 128)
conv (4, 256, 2), LReLU	(16 x 16 x 256)
conv (4, 512, 2), LReLU	(8 x 8 x 512)
conv (4, 1024, 2), LReLU	(4 x 4 x 1024)
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
Linear (200), LReLU	200
Linear (200), LReLU	200
Linear (200), LReLU	200
Linear (200), LReLU	200
Linear (200), LReLU	200
Linear (1)	1
EBM $f_{\alpha_2}(\mathbf{z}_2)$	
Input: $\mathbf{z}_2$	100
Linear (100), LReLU	100
Linear (100), LReLU	100
Linear (1)	1

Table 5. Network structures for generation, inference and EBMs on CELEBA-64 and EBM structure for NVAE backbone models.

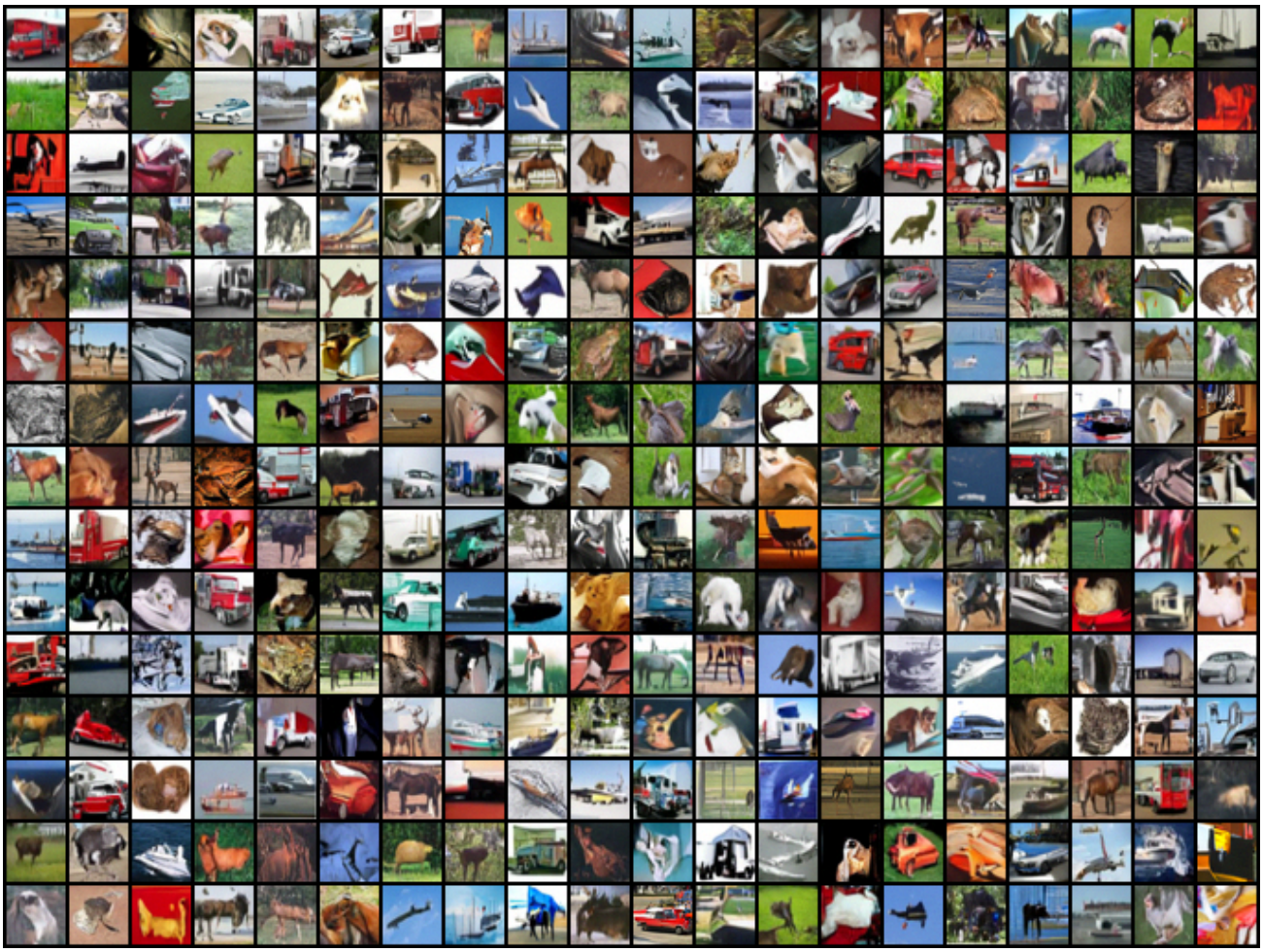


Figure 6. Generated images on CIFAR-10. Samples are uncurated.

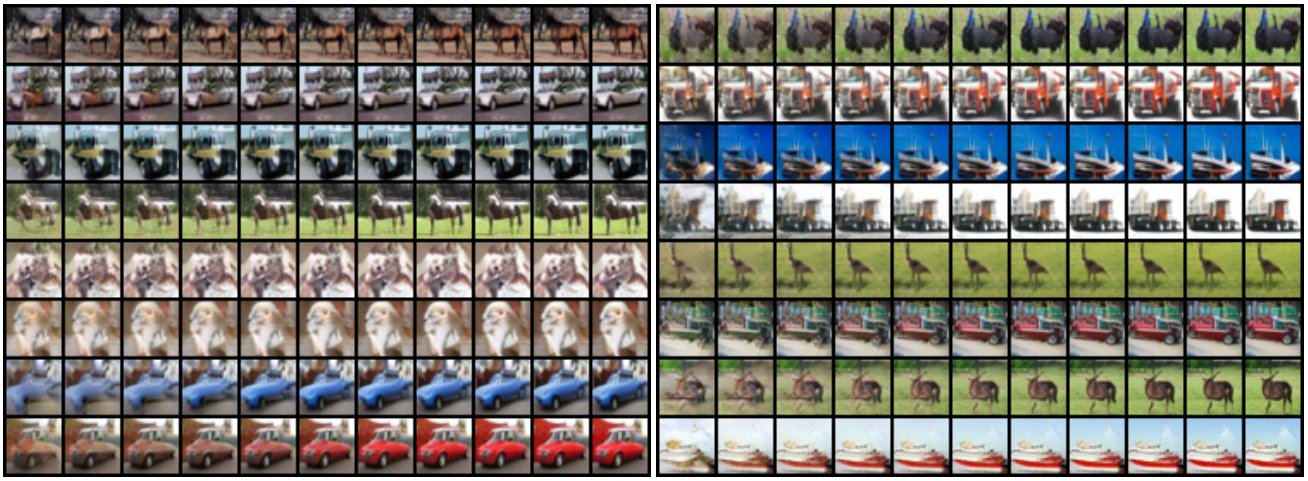


Figure 7. Langevin transition on CIFAR-10.

Figure 8. Generated images on LSUN-Church-64. Samples are uncurated.

Figure 9. Langevin transition on LSUN-Church-64.

Figure 10. Generated images on CelebA-HQ-256 (temperature t=0.7). Samples are uncurated.

Figure 11. Generated images on CelebA-HQ-256 (temperature t=1.0). Samples are uncurated.

References

[1] Jyoti Aneja, Alex Schwing, Jan Kautz, and Arash Vahdat. A contrastive learning approach for training variational autoencoder priors. *Advances in neural information processing systems*, 34:480–493, 2021. 3

[2] Jakob D Drachmann Havtorn, Jes Frellsen, Soren Hauberg, and Lars Maaløe. Hierarchical vaes know what they don't know. In *International Conference on Machine Learning*, pages 4117–4128. PMLR, 2021. 3, 4

[3] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 1

[4] 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. 2, 3

[5] 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. 4

[6] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *Advances in Neural Information Processing Systems*, 33:19667–19679, 2020. 2

[7] 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, 3