VAE + EBM = VAEBM: Taking the Best of Both Worlds

Haozhen Shen
1003115112
haozhen.shen@mail.utoronto.ca

Peifeng Zhu
1007036592
peifeng.zhu@mail.utoronto.ca

Bingchen Yang
1005894613
bingchen.yang@mail.utoronto.ca

Abstract

Variational autoencoders (VAEs), and energy-based models (EBMs) are competing likelihood models for deep generative learning. VAEs have the advantage of fast sample generation and is more interpretable due to its tractable latent space. However, VAEs usually suffer from mode collapse. It tend to assigning high probability density to regions outside of the actual distribution and failing to generate sharp images. Recently more attention have been draw on energy-based models due to their flexibility and generality. However, the unknown normalizing constant of EBMs are intractable, and sampling from them requires Markov chain Monte Carlo (MCMC) iterations that mix slowly in high dimensional pixel space. We focus on the method namely VAEBM. which consists of a product of a VAE generator and an EBM component defined in the pixel space which alleviates the problems mentioned above in both models and result in a more powerful model with the two combined. In this paper we provide a example of VAEBM and explain how does it take the best from both worlds with illustrations on the CIFAR10 dataset.[Xiao et al. 2021]

1 Introduction

Generative models have draw much interest in the field of machine learning in the past decade. They aim to capture the inner probabilistic distribution of a class of data, and therefore able to generate similar data. It has found diverse applications in numerous fields and problems such as image synthesis [Huang et al. 2018], speech recognition and generation [van den Oord et al. 2016a], natural language processing [Calixto et al. 2019], [He et al. 2019]. Competing frameworks for likelihood-based models include variational autoencoders (VAEs) [Kingma and Welling 2014], normalizing flows [Rezende and Mohamed 2016] [Dinh et al. 2017], autoregressive models [van den Oord et al. 2016b], and energy-based models (EBMs) [Du and Mordatch 2020]. Such models can often be trained in a stabilized fashion by maximizing the data likelihood under the model. Other competing models such as generative adversarial networks (GANs) [Goodfellow et al. 2014] are instead trained with loss objectives that plays a zero sum game between two agents, and are usually unstable in training.

In this paper, we mainly focus on VAEs and EBMs and finally the combination of the two VAEBM[Xiao et al. 2021]. VAEs have the advantage of fast sample generation, easy-to-access encoding networks, and is more interpretable due to its easily traversable latent space. However, without carefully designed latent spaces and network architectures[Vahdat and Kautz 2021] they are usually outperformed by most other models such as normalizing flows and autoregressive models.

Preprint. Under review.

Models with a tractable likelihood are in general constrained. Such models assume an exact synthesis of pseudo-data from the model can be done with a specified, tractable procedure. In the setting of VAEs, the data is modeled as a directed latent-variable model. These assumptions might not always be natural.EBMs model the data density in a unnormalized fashion by assigning low energy to high-probability regions in the data space. Such simple modeling mechanism makes them require almost no restrictions on network architectures, and are potentially more flexible and expressive. However, training EBMs requires computing the intractable normalizing constant which is usually estimated by sampling form the distribution through MCMC. Such iterations can suffer from slow mode mixing and is computationally expensive when neural networks represent the energy function.

VAEBM is the composition of a VAE and EBM, it generative distribution of VAEBM is defined as the product of a VAE generator and an EBM component defined in pixel space.

"Intuitively, the VAE captures the majority of the mode structure in the data distribution. However, it may still generate samples from low-probability regions in the data space. Thus, the energy function focuses on refining the details and reducing the likelihood of non-data-like regions, which leads to significantly improved samples." —Xiao et al. 2021

The training of VAEBM can be separated by first training the VAE component, then training the EBM component to refine the data distribution captured by the VAE. Analogous to GANs, the EBM component acts like a discriminator in the GAN setup. As VAEs suffer from mode collapse, an EBM can help reduce the likelihood of non-data-like regions and thus cover more modes of the data distribution. We illustrate how this is done using the CIFAR10 data set, where first the images generated by the VAE are in general blurry but after refining by the EBM component, we were able to obtain sharp and crisp images.

Our goal in this paper are threefold. Firstly, we present the downfalls of simple modeling using either a VAE or EBM for CIFAR10. Secondly, we highlight the advantages of the VAEBM model in terms of training procedure and model generalization with implicit image generation. Finally, we show that VAEBM is a effective composition of the two and can not only generate high quality images, but are also useful on tasks such as out-of-distribution generalization.

Related Work: VAEBM provided a symbiotic composition of an EBM with a VAE. Some other work also tried to combine the two but in a different manner. [Pang et al. 2020] use EBMs for the prior distribution which has the advantage of fast mixing time due to the low dimensionality of the latent space. [Han et al. 2020] proposed a a joint training method to learn both the VAE and the latent EBM, their objective function is of an symmetric and anti-symmetric form of divergence triangle that integrates variational and adversarial learning. Other methods also tries to incorporate an EBM with other models, such as [Che et al. 2020] which formulates GANs into an EBM perspective,

2 Model Components

Variational Auto Encoders (VAEs): In the VAEs setup, the main goal is to train a generative model of the form $p(x, z) = p(z)p(x|z)$ where $p(z)$ is a prior distribution over latent variables z and $p(x|z)$ is the likelihood function conditioned on the latent variables. In other words the decoder that generates data x given latent variables z. In most cases the true posterior $p(z|x)$ is intractable, thus we make use of an approximate posterior distribution $q_\phi(z|x)$ to train the generative model where the variational lower bound on $\log p_\theta(x)$ is maximized with $q_\phi(z|x)$ as the approximate posterior:

$$\log p_\theta(\mathbf{x}) \geq \mathbb{E}_{z\sim q_\psi(z|x)}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{KL}(q_\psi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))] :=:= \mathcal{L}_{vae}(\mathbf{x}, \theta, \phi) \tag{1}$$

Energy Based Models (EBMs): For an inpout x, define $E_\theta(\mathbf{x}) \in \mathbb{R}$ to be the energy function. This function is represented by a deep neural network with parameters θ. The energy function defines a probability distribution via the Boltzmann distribution $p_\theta(\mathbf{x}) = \frac{1}{Z(\theta)} exp(E_\theta(\mathbf{x}))$ where $Z(\theta)$ the unknown normalizing constant denotes the partition function.

For a set of samples drawn from the data distribution $p_d(\mathbf{x})$, the goal of maximum likelihood learning is to maximize the log-likelihood $L(\psi) = E_{x\sim p_d(\mathbf{x})}[\log p_\psi(\mathbf{x})]$, where

$$\partial_\psi L(\psi) = \mathbb{E}_{\mathbf{x}\sim p_d(\mathbf{x})}[-\partial_\psi E_\psi(\mathbf{x})] + \mathbb{E}_{\mathbf{x}\sim p_\psi(\mathbf{x})}[\partial_\psi E_\psi(\mathbf{x})] \tag{2}$$

In order to generate samples from this distribution, previous work mainly relies on MCMC methods such as random walk and Gibbs sampling. These methods usually suffer from slow mode mixing especially for high-dimensional image data. To improve the sampling procedure, Langevin dynamics [Neal 1993] which makes use of the gradient of the energy function to undergo sampling. With initial sample x_0,

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \frac{\eta}{2}\nabla_x E_\psi(\mathbf{x}_t) + \sqrt{\eta}\omega t, \qquad \omega_t \sim N(0, I) \tag{3}$$

where η is the step-size. This procedure is normally run for finite iteration which yield a markov chain to generates samples from the distribution defined by the energy function.

3 ENERGY-BASED VARIATIONAL AUTOENCODERS (VAEBMs)

The VAE has the ability to capture the majority of the modes in the data distribution. Whereas the energy function focuses on refining the details and reducing the likelihood of non-data-like regions, thus leading to significantly improved samples. The VAEBM generator is defined as,

$$h_{\psi,\theta}(\mathbf{x}, \mathbf{z}) = \frac{1}{Z_{\psi,\theta}} p_\theta(\mathbf{x}, \mathbf{z}) e^{-E_\psi(x)} \tag{4}$$

Where $p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z})$ is the VAE generator, $E_\psi(\mathbf{x})$ is a neural network-based energy function, operating only in the x space, and $Z_{\psi,\theta} = \int p_\theta(\mathbf{x}) e^{E_\psi(\mathbf{x})} d\mathbf{x}$ is the normalization constant. Laterψ and θ, are trained by maximizing the marginal log-likelihood on the training data.

$$\log h_{\psi,\theta}(\mathbf{x}, \mathbf{z}) = \log p_\theta(\mathbf{x}, \mathbf{z}) - E_\psi(\mathbf{x}) - \log Z_{\psi,\theta} \tag{5}$$

$$\geq \underbrace{E_{z\sim q_\psi(z|x)}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{KL}(q_\psi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))}_{\mathcal{L}_{vae}(x,\theta,\phi)} - \underbrace{E_\psi(\mathbf{x}) - \log Z_{\psi,\theta}}_{\mathcal{L}_{EBM}(x,\psi,\theta)} \tag{6}$$

Where the variational lower bound is replaced in to train the objective. The first term corresponds to the VAE objective and the second term corresponds to training the EBM component.

4 Training

Taking a look of our combined objective Eq.6 we see that the first part $\mathcal{L}_{vae}(\mathbf{x}, \theta, \phi)$ is identical to the vanilla VAE's objective. However, the log partition $\log Z_{\psi,\theta}$ in the second part depends on both ψ and θ. This is problematic since applying a similar training procedure for a EBM will no longer work. We take a deeper look on how this leads differently in training. Observe that the log partition $\log Z_{\psi,\theta}$ has gradients,

$$\partial_\psi \log Z_{\psi,\theta} = \mathbb{E}_{\mathbf{x}\sim h_{\psi,\theta}(\mathbf{x},\mathbf{z})}[-\partial_\psi E_\psi(\mathbf{x})], \qquad \partial_\theta \log Z_{\psi,\theta} = \mathbb{E}_{\mathbf{x}\sim p_\psi(x)}[\partial_\theta \log p_\theta(\mathbf{x})] \tag{7}$$

The gradient with respect to ψ can be easily estimated by evaluating the gradient of the energy function at samples drawn from the VAEBM model $h_{\psi,\theta}(\mathbf{x}, \mathbf{x})$ using MCMC. However, the gradient with respect to θ involves the intractable $\partial_\theta \log p_\theta(\mathbf{x})$. Xiao shown that this requires sampling from the VAE's posterior distribution $\mathbf{x} \sim h_{\psi,\theta}(\mathbf{x}, \mathbf{z})$ which drastically increases computational complexity. To avoid such computational complexity an alternative is proposed by Xiao, specifically we can divide the training procedure into two stages where in the first stage we train our VAE component with the traditional $\mathcal{L}_{vae}(\mathbf{x}, \theta, \phi)$ objective. Note that this objective is identical to the vanilla VAE's objective. In the second stage we fix our VAE component, in other words we fix our parameter θ and only train the EBVM component. The resulting derivative of the EBM component with respect to ψ becomes,

$$\partial_\psi \mathcal{L}(\psi) = \mathbb{E}_{\mathbf{x}\sim p_d(\mathbf{x})}[-\partial_\psi E_\psi(\mathbf{x})] + \mathbb{E}_{\mathbf{x}\sim h_{\psi,\theta}(\mathbf{x},\mathbf{z})}[\partial_\psi E_\psi(\mathbf{x})], \tag{8}$$

Which is then equivalent to our EBM's objective described in Eq.2.

Sampling using Langevin Dynamics for EBM component: The sampling procedure differs from the traditional Langevin Dynamics sampling for EBMs. We now sample from the joint space of

$\mathbf{x}$ and $\mathbf{z}$, and further more Xiao proposed that we can accelerate the sampling procedure use re-parametrization, Consider a Gaussian distribution as a simple example: if $\mathbf{z} \sim \mathcal{N}(\mu_{\mathbf{z}}, \sigma_{\mathbf{z}})$ and $\mathbf{x}|\mathbf{z} \sim \mathcal{N}(\mu_{\mathbf{x}}(\mathbf{z}), \sigma_{\mathbf{x}}(\mathbf{z}))$, then

$$\mathbf{z} = T_\theta^{\mathbf{z}}(\boldsymbol{\epsilon}_{\mathbf{z}}) = \mu_{\mathbf{z}} + \sigma_{\mathbf{z}} \cdot \boldsymbol{\epsilon}_{\mathbf{z}}, \quad \mathbf{x} = T_\theta^{\mathbf{x}}(\boldsymbol{\epsilon}_{\mathbf{x}}, \boldsymbol{\epsilon}_{\mathbf{z}}) = \mu_{\mathbf{x}}(\mathbf{z}) + \sigma_{\mathbf{x}}(\mathbf{z}) \cdot \boldsymbol{\epsilon}_{\mathbf{x}},$$

and

$$J_{T_\theta^{-1}}(\mathbf{x}, \mathbf{z}) = [\sigma_{\mathbf{x}}(\mathbf{z})^{-1}, \sigma_{\mathbf{z}}^{-1}].$$

Recall that the generative model of our EBM is

$$h_{\psi,\theta}(\mathbf{x}, \mathbf{z}) = \frac{e^{-E_\psi(\mathbf{x})} p_\theta(\mathbf{x}, \mathbf{z})}{Z_{\psi,\theta}}. \tag{9}$$

We can apply the change of variable to $h_{\psi,\theta}(\mathbf{x}, \mathbf{z})$ in similar manner:

$$h_{\psi,\theta}(\boldsymbol{\epsilon}_{\mathbf{x}}, \boldsymbol{\epsilon}_{\mathbf{z}}) = h_{\psi,\theta}(T_\theta(\boldsymbol{\epsilon}_{\mathbf{x}}, \boldsymbol{\epsilon}_{\mathbf{z}})) \left|\det\left(J_{T_\theta}(\boldsymbol{\epsilon}_{\mathbf{x}}, \boldsymbol{\epsilon}_{\mathbf{z}})\right)\right|, \tag{10}$$

where J_{T_θ} is the Jacobian of T_θ.

Since we have the relation

$$J_{\mathbf{f}^{-1}} \circ \mathbf{f} = J_{\mathbf{f}}^{-1} \tag{11}$$

for invertible function $\mathbf{f}$, we have that

$$h_{\psi,\theta}(\boldsymbol{\epsilon}_{\mathbf{x}}, \boldsymbol{\epsilon}_{\mathbf{z}}) = \frac{1}{Z_{\psi,\theta}} e^{-E_\psi(T_\theta(\boldsymbol{\epsilon}_{\mathbf{x}}, \boldsymbol{\epsilon}_{\mathbf{z}}))} p_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}_{\mathbf{x}}, \boldsymbol{\epsilon}_{\mathbf{z}}), \tag{12}$$

After we obtained samples $(\boldsymbol{\epsilon}_{\mathbf{x}}, \boldsymbol{\epsilon}_{\mathbf{z}})$ from the distribution, we obtain $(\mathbf{x}, \mathbf{z})$ by applying the transformation T_θ. Note that this sampling scheme is extremely useful when data is encoded to a high dimensional latent space. Since we reparameterized the variables into the same scale we do not need to tune the sampling scheme for each variable. For example when a hierarchical variation autoencoder is used as the VAE component.

Advantages: Note that the combined model resolves some of the problems in a single VAE or EBM. We expect our model to address the problem of mode collapse in the VAE setup, as well as reducing the number of MCMC iterations required for mode mixing compared to a single EBM. Moreover, the combined model avoids the difficulties in estimating the full gradient of the normalizing constant $logZ_{\psi,\theta}$ by deploying a two stage training. This has additional advantages. As in the first stage we minimize the distance between the VAE model and the data distribution, our pretrained VAE will obtain a good approximation to the data distribution. Therefore, in the second stage we expect our EBM to reduce the mismatch between the model and the data distribution with a relatively small number of expensive updates.

5 Experiments

In this section, we evaluate our smaller example of VAEBM through comprehensive experiments on the dataset CIFAR10. We provide some ablation studies of model architecture and study mode coverage of our model. We draw approximate samples both for training and testing by running short Langevin dynamics chains on the distribution. We use the exact architecture of the EBM component described in original paper.

Deviations from the Original Paper: Due to limiting computing resource we did not use NVAE [Vahdat and Kautz 2021] as the VAE component. We instead used the ResNet18 architecture for both the encoder and decoder following the official implementation in Pytorch Lightning.

5.1 Image Generation

In this section we qualitatively compare some of the images generated by our VAEBM to images generated by a single VAE and images generated by a single EBM. We show that such composition of the two indeed enhances the model and provides better results.

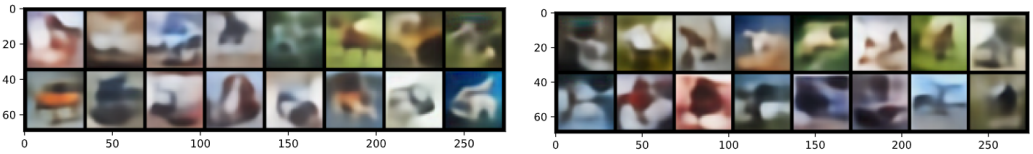

Figure 1: CIFAR10 Samples Generated by Pretrained VAE

We trained our VAE for 50 epochs with learning rate $1e-4$ and batch size of 32 using Adam optimizer. Our training loss converges to a small number. Although our training reaches a significant low loss our VAE still cannot generate sharp images. We also trained a vanilla EBM for 20 epochs with with learning rate $1e-4$ and batch size of 32 using Adam optimizer. Since each epoch take half an hour and training loss drops slowly, we early stopped training. Although training might be insufficient We observe that it take a significant amount of MCMC iteration before mixing of modes. We illustrate some of the generated image from both models.

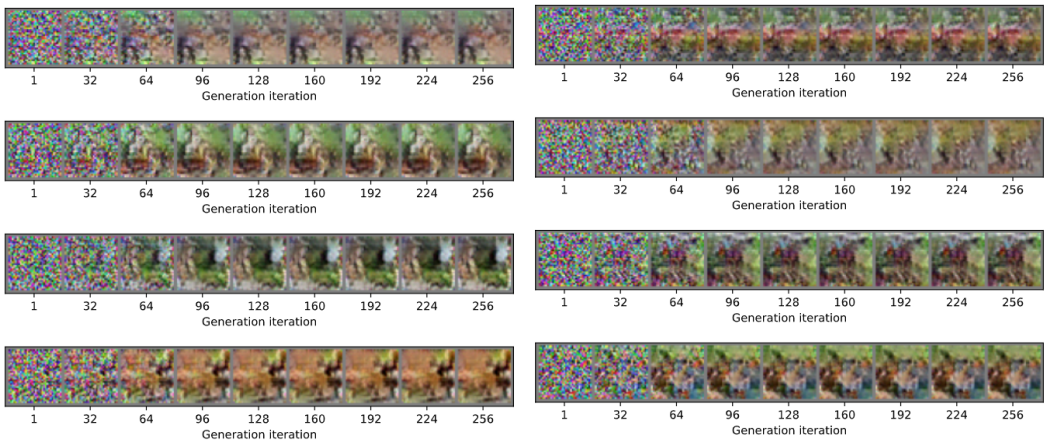

Figure 2: CIFAR10 Samples Generated by EBM Along MCMC Chain

Finally we trained our VAEBM for 20 epochs with learning rate $1e-4$ and batch size of 32 using Adam optimizer. We illustrate some of our results generated by our VAEBM.

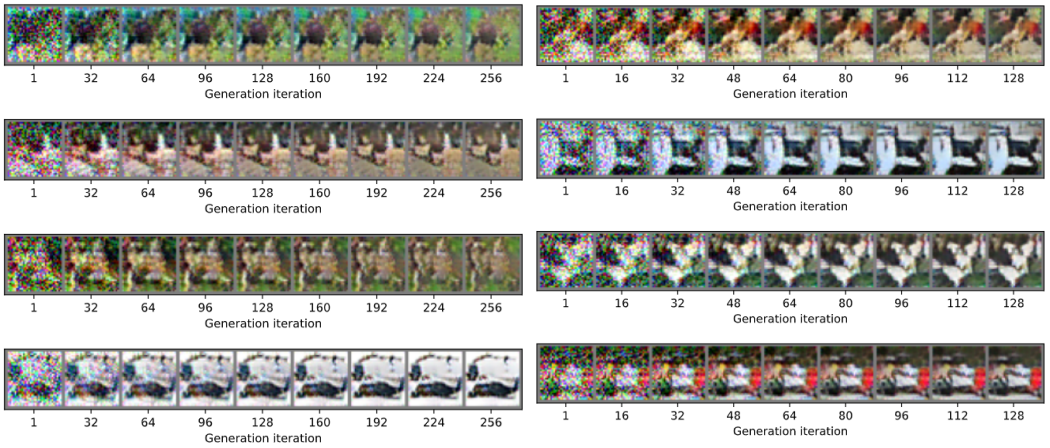

Figure 3: CIFAR10 Samples Generated by VAEBM Along MCMC Chain

5.2 Comparison

Comparing the generated images we see that our VAEBM has the best quality of generated images among the three. Compare to the VAE and EBM, our VAEBM was able to produce sharp images within a small number of MCMC iterations.

6 Out of Distribution Detection

We illustrate that our VAEBM has the ability to detect out-of-distribution data in Fig.4, sometimes referred to as "anomaly" detection). We inject random noise to some of the test images of CIFAR10 and compare energy levels produced by our model with the original image. A lower output of the model denotes a low probability. Thus, we hope to see low scores if we inject random noise to the model.

Figure 4: Energy levels compare by injecting noise to test set

7 Conclusion

We demonstrated a simple version of VAEBM to illustrate a symbiotic composition of an energy-based generative model with an variational autoencoder. The data distribution of our model is defined jointly by a VAE and an energy network, the EBM component of the model. We illustrated how both models benefits from each other through this combination, the VAE component benefits from being able to overcome mode collapse to some degree, where the EBM component benefits from being able to mix faster during MCMC iterations. We illustrated how the training procedure and sampling procedure of the model is carefully derived and implemented. And through comprehensive experiments we were able to illustrate the effectiveness of VAEBM in image generation, and also introduces some other appealing properties of the model such as being able to distinguish out of distribution data. Future work might include applying model to larger data sets, transitioning the EBM component to the latent space, or applying more advanced sampling techniques.

References

I. Calixto, M. Rios, and W. Aziz. Latent variable model for multi-modal translation, 2019.

T. Che, R. Zhang, J. Sohl-Dickstein, H. Larochelle, L. Paull, Y. Cao, and Y. Bengio. Your gan is secretly an energy-based model and you should use discriminator driven latent sampling, 2020.

L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using real nvp, 2017.

Y. Du and I. Mordatch. Implicit generation and generalization in energy-based models, 2020.

I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks, 2014.

T. Han, E. Nijkamp, L. Zhou, B. Pang, S.-C. Zhu, and Y. N. Wu. Joint training of variational auto-encoder and latent energy-based model, 2020.

J. He, D. Spokoyny, G. Neubig, and T. Berg-Kirkpatrick. Lagging inference networks and posterior collapse in variational autoencoders, 2019.

H. Huang, P. S. Yu, and C. Wang. An introduction to image synthesis with generative adversarial nets, 2018.

D. P. Kingma and M. Welling. Auto-encoding variational bayes, 2014.

R. M. Neal. *Probabilistic inference using Markov chain Monte Carlo methods*. 1993.

B. Pang, T. Han, E. Nijkamp, S.-C. Zhu, and Y. N. Wu. Learning latent space energy-based prior model, 2020.

D. J. Rezende and S. Mohamed. Variational inference with normalizing flows, 2016.

A. Vahdat and J. Kautz. Nvae: A deep hierarchical variational autoencoder, 2021.

A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio, 2016a.

A. van den Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu. Conditional image generation with pixelcnn decoders, 2016b.

Z. Xiao, K. Kreis, J. Kautz, and A. Vahdat. Vaebm: A symbiosis between variational autoencoders and energy-based models, 2021.