Implicit Generation and Modeling with Energy-Based Models

Yilun Du *
MIT CSAIL

Igor Mordatch
Google Brain

Abstract

Energy based models (EBMs) are appealing due to their generality and simplicity in likelihood modeling, but have been traditionally difficult to train. We present techniques to scale MCMC based EBM training on continuous neural networks, and we show its success on the high-dimensional data domains of ImageNet32x32, ImageNet128x128, CIFAR-10, and robotic hand trajectories, achieving better samples than other likelihood models and nearing the performance of contemporary GAN approaches, while covering all modes of the data. We highlight some unique capabilities of implicit generation such as compositionality and corrupt image reconstruction and inpainting. Finally, we show that EBMs are useful models across a wide variety of tasks, achieving state-of-the-art out-of-distribution classification, adversarially robust classification, state-of-the-art continual online class learning, and coherent long term predicted trajectory rollouts.

1 Introduction

Learning models of the data distribution and generating samples are important problems in machine learning for which a number of methods have been proposed, such as Variational Autoencoders (VAEs) [Kingma and Welling, 2014] and Generative Adversarial Networks (GANs) [Goodfellow et al., 2014].In this work, we advocate for continuous energy-based models (EBMs), represented as neural networks, for generative modeling tasks and as a building block for a wide variety of tasks. These models aim to learn an energy function $E(\mathbf{x})$ that assigns low energy values to inputs $\mathbf{x}$ in the data distribution and high energy values to other inputs. Importantly, they allow the use of an *implicit* sample generation procedure, where sample $\mathbf{x}$ is found from $\mathbf{x} \sim e^{-E(\mathbf{x})}$ through MCMC sampling. Combining implicit sampling with energy-based models for generative modeling has a number of conceptual advantages compared to methods such as VAEs and GANs which use explicit functions to generate samples:

Simplicity and Stability: An EBM is the only object that needs to be trained and designed. Separate networks are not tuned to ensure balance (for example, [He et al., 2019] point out unbalanced training can result in posterior collapse in VAEs or poor performance in GANs [Kurach et al., 2018]).

Sharing of Statistical Strength: Since the EBM is the only trained object, it requires fewer model parameters than approaches that use multiple networks. More importantly, the model being concentrated in a single network allows the training process to develop a shared set of features as opposed to developing them redundantly in separate networks.

Adaptive Computation Time: Implicit sample generation in our work is an iterative stochastic optimization process, which allows for a trade-off between generation quality and computation time.

*Work done at OpenAI
*Correspondence to: yilundu@mit.edu
*Additional results, source code, and pre-trained models are available at https://sites.google.com/view/igebm

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

This allows for a system that can make fast coarse guesses or more deliberate inferences by running the optimization process longer. It also allows for refinement of external guesses.

Flexibility Of Generation: The power of an explicit generator network can become a bottleneck on the generation quality. For example, VAEs and flow-based models are bound by the manifold structure of the prior distribution and consequently have issues modeling discontinuous data manifolds, often assigning probability mass to areas unwarranted by the data. EBMs avoid this issue by directly modeling particular regions as high or lower energy.

Compositionality: If we think of energy functions as costs for a certain goals or constraints, summation of two or more energies corresponds to satisfying all their goals or constraints [Mnih and Hinton, 2004, Haarnoja et al., 2017]. While such composition is simple for energy functions (or product of experts [Hinton, 1999]), it induces complex changes to the generator that may be difficult to capture with explicit generator networks.

Despite these advantages, energy-based models with implicit generation have been difficult to use on complex high-dimensional data domains. In this work, we use Langevin dynamics [Welling and Teh, 2011], which uses gradient information for effective sampling and initializes chains from random noise for more mixing. We further maintain a replay buffer of past samples (similarly to [Tieleman, 2008] or [Mnih et al., 2013]) and use them to initialize Langevin dynamics to allow mixing between chains. An overview of our approach is presented in Figure 1.

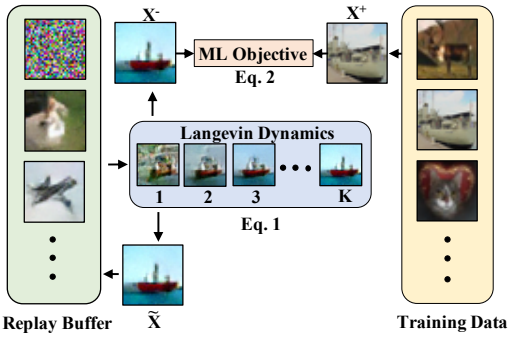


Figure 1: Overview of our method and the interrelationship of the components involved.

Empirically, we show that energy-based models trained on CIFAR-10 or ImageNet image datasets generate higher quality image samples than likelihood models and nearing that of contemporary GANs approaches, while not suffering from mode collapse. The models exhibit properties such as correctly assigning lower likelihood to out-of-distribution images than other methods (no spurious modes) and generating diverse plausible image completions (covering all data modes). Implicit generation allows our models to naturally denoise or inpaint corrupted images, convert general images to an image from a specific class, and generate samples that are compositions of multiple independent models.

Our contributions in this work are threefold. Firstly, we present an algorithm and techniques for training energy-based models that scale to challenging high-dimensional domains. Secondly, we highlight unique properties of energy-based models with implicit generation, such as compositionality and automatic decorruption and inpainting. Finally, we show that energy-based models are useful across a series of domains, on tasks such as out-of-distribution generalization, adversarially robust classification, multi-step trajectory prediction and online learning.

2 Related Work

Modeling data using the Boltzmann distribution has been used extensively across diverse fields. Such models include Ising models [Cipra, 1987] in electromagnetism, Markov Logic Networks [Richardson and Domingos, 2006] over knowledge bases, the Helmholtz [Dayan et al., 1995] and Boltzmann Machines [Ackley et al., 1985] in machine learning, and the FRAME model [Zhu et al., 1998] in computer vision.

In the computer vision community, the FRAME model [Zhu et al., 1998, Wu et al., 2000] utilizes the Boltzmann distribution or the Gibbs distribution to represent different textures of images, as well as a model of texture perception. The FRAME model is further extended for modeling object patterns in [Xie et al., 2015, 2016b] Lu et al. [2016] uses deep network features in the FRAME model for image synthesis. Xie et al. [2016c] further extends the FRAME model to the energy-based generative ConvNet, in which the energy function is parameterized by a ConvNet structure. A multi-grid sampling and training strategy for the energy-based generative ConvNet is also studied in [Gao et al., 2018].

In the deep learning community, such models are are known as Energy-based models (EBMs). Ackley et al. [1985], Hinton [2006], Salakhutdinov and Hinton [2009] proposed latent based EBMs where energy is represented as a composition of latent and observable variables. In contrast Mnih and Hinton [2004], Hinton et al. [2006] proposed EBMs where inputs are directly mapped to outputs, a structure we follow. We refer readers to [LeCun et al., 2006] for a comprehensive tutorial on energy models.

The primary difficulty in training EBMs comes from effectively estimating and sampling the partition function. One approach to train energy based models is to sample the partition function through amortized generation. Kim and Bengio [2016], Zhao et al. [2016], Haarnoja et al. [2017], Kumar et al. [2019] propose learning a separate network to generate samples, while Xie et al. [2016a, 2018] use a separate network to initialize MCMC sampling, which makes these methods closely connected to GANs [Finn et al., 2016], but these methods do not have the advantages of implicit sampling noted in the introduction. Furthermore, amortized generation is prone to mode collapse, especially when training the sampling network without an entropy term which is often approximated or ignored.

An alternative approach is to use MCMC sampling to directly estimate the partition function. This has an advantage of provable mode exploration and allows the benefits of implicit generation listed in the introduction. Hinton [2006] proposed Contrastive Divergence, which uses gradient free MCMC chains initialized from training data to estimate the partition function. Similarly, Salakhutdinov and Hinton [2009] apply contrastive divergence, while Tieleman [2008] proposes PCD, which propagates MCMC chains throughout training. By contrast, we initialize chains from random noise, allowing each mode of the model to be visited with equal probability. But initialization from random noise comes at a cost of longer mixing times. As a result we use Gradient based MCMC (Langevin Dynamics) for more efficient sampling and to offset the increase of mixing time which was also studied previously in [Teh et al., 2003, Xie et al., 2016c]. We note that HMC [Neal, 2011] may be an even more efficient gradient algorithm for MCMC sampling, though we found Langevin Dynamics to be more stable. To allow gradient based MCMC, we use continuous inputs, while most approaches have used discrete inputs. We build on idea of PCD and maintain a replay buffer of past samples to additionally reduce mixing times.

3 Energy-Based Models and Sampling

Given a datapoint $\mathbf{x}$, let $E_\theta(\mathbf{x}) \in \mathbb{R}$ be the energy function. In our work this function is represented by a deep neural network parameterized by weights θ. The energy function defines a probability distribution via the Boltzmann distribution $p_\theta(\mathbf{x}) = \frac{\exp(-E_\theta(\mathbf{x}))}{Z(\theta)}$, where $Z(\theta) = \int \exp(-E_\theta(\mathbf{x}))d\mathbf{x}$ denotes the partition function. Generating samples from this distribution is challenging, with previous work relying on MCMC methods such as random walk or Gibbs sampling [Hinton, 2006]. These methods have long mixing times, especially for high-dimensional complex data such as images. To improve the mixing time of the sampling procedure, we use Langevin dynamics which makes use of the gradient of the energy function to undergo sampling

$$\tilde{\mathbf{x}}^k = \tilde{\mathbf{x}}^{k-1} - \frac{\lambda}{2}\nabla_{\mathbf{x}} E_\theta(\tilde{\mathbf{x}}^{k\check{}1}) + \omega^k,\ \omega^k \sim \mathcal{N}(0, \lambda) \tag{1}$$

where we let the above iterative procedure define a distribution q_θ such that $\tilde{\mathbf{x}}^K \sim q_\theta$. As shown by Welling and Teh [2011] as $K \to \infty$ and $\lambda \to 0$ then $q_\theta \to p_\theta$ and this procedure generates samples from the distribution defined by the energy function. Thus, samples are generated implicitly[†] by the energy function E as opposed to being explicitly generated by a feedforward network.

In the domain of images, if the energy network has a convolutional architecture, energy gradient $\nabla_{\mathbf{x}} E$ in (1) conveniently has a deconvolutional architecture. Thus it mirrors a typical image generator network architecture, but without it needing to be explicitly designed or balanced. We take two views of the energy function E: firstly, it is an object that defines a probability distribution over data and secondly it defines an implicit generator via (1).

3.1 Maximum Likelihood Training

We want the distribution defined by E to model the data distribution p_D, which we do by minimizing the negative log likelihood of the data $\mathcal{L}_{\text{ML}}(\theta) = \mathbb{E}_{\mathbf{x}\sim p_D}\left[-\log p_\theta(\mathbf{x})\right]$ where $-\log p_\theta(\mathbf{x}) = E_\theta(\mathbf{x}) -$

[†]Deterministic case of procedure in (1) is $\mathbf{x} = \arg\min E(\mathbf{x})$, which makes connection to implicit functions more clear.

$\log Z(\theta)$. This objective is known to have the gradient (see [Turner, 2005] for derivation) $\nabla_\theta \mathcal{L}_{\text{ML}} = \mathbb{E}_{\mathbf{x}^+ \sim p_D}\left[\nabla_\theta E_\theta(\mathbf{x}^+)\right] - \mathbb{E}_{\mathbf{x}^- \sim p_\theta}\left[\nabla_\theta E_\theta(\mathbf{x}^-)\right]$. Intuitively, this gradient decreases energy of the positive data samples $\mathbf{x}^+$, while increasing the energy of the negative samples $\mathbf{x}^-$ from the model p_θ. We rely on Langevin dynamics in (1) to generate q_θ as an approximation of p_θ:

$$\nabla_\theta \mathcal{L}_{\text{ML}} \approx \mathbb{E}_{\mathbf{x}^+ \sim p_D}\left[\nabla_\theta E_\theta(\mathbf{x}^+)\right] - \mathbb{E}_{\mathbf{x}^- \sim q_\theta}\left[\nabla_\theta E_\theta(\mathbf{x}^-)\right]. \tag{2}$$

This is similar to the gradient of the Wasserstein GAN objective [Arjovsky et al., 2017], but with an implicit MCMC generating procedure and no gradient through sampling. This lack of gradient is important as it controls between the diversity in likelihood models and the mode collapse in GANs.

The approximation in (2) is exact when Langevin dynamics generates samples from p, which happens after a sufficient number of steps (mixing time). We show in the supplement that p_d and q appear to match each other in distribution, showing evidence that p matches q. We note that even in cases when a particular chain does not fully mix, since our initial proposal distribution is a uniform distribution, all modes are still equally likely to be explored.

3.2 Sample Replay Buffer

Langevin dynamics does not place restrictions on sample initialization $\tilde{\mathbf{x}}^0$ given sufficient sampling steps. However initialization plays an crucial role in mixing time. Persistent Contrastive Divergence (PCD) [Tieleman, 2008] maintains a single persistent chain to improve mixing and sample quality. We use a sample replay buffer $\mathcal{B}$ in which we store past generated samples $\tilde{\mathbf{x}}$ and use either these samples or uniform noise to initialize Langevin dynamics procedure. This has the benefit of continuing to refine past samples, further increasing number of sampling steps K as well as sample diversity. In all our experiments, we sample from $\mathcal{B}$ 95% of the time and from uniform noise otherwise.

3.3 Regularization and Algorithm

Arbitrary energy models can have sharp changes in gradients that can make sampling with Langevin dynamics unstable. We found that constraining the Lipschitz constant of the energy network can ameliorate these issues. To constrain the Lipschitz constant, we follow the method of [Miyato et al., 2018] and add spectral normalization to all layers of the model. Additionally, we found it useful to weakly L2 regularize energy magnitudes for both positive and negative samples during training, as otherwise while the difference between positive and negative samples was preserved, the actual values would fluctuate to numerically unstable values. Both forms of regularization also serve to ensure that partition function is integrable over the domain of the input, with spectral normalization ensuring smoothness and L2 coefficient bounding the magnitude of the unnormalized distribution. We present the algorithm below, where $\Omega(\cdot)$ indicates the stop gradient operator.

Algorithm 1 Energy training algorithm

Input: data dist. $p_D(\mathbf{x})$, step size λ, number of steps K
$\mathcal{B} \leftarrow \varnothing$
while not converged **do**
  $\mathbf{x}_i^+ \sim p_D$
  $\mathbf{x}_i^0 \sim \mathcal{B}$ with 95% probability and $\mathcal{U}$ otherwise
  ▷ *Generate sample from q_θ via Langevin dynamics:*
  for sample step $k = 1$ to K **do**
    $\tilde{\mathbf{x}}^k \leftarrow \tilde{\mathbf{x}}^{k-1} - \nabla_{\mathbf{x}} E_\theta(\tilde{\mathbf{x}}^{k-1}) + \omega, \quad \omega \sim \mathcal{N}(0, \sigma)$
  end for
  $\mathbf{x}_i^- = \Omega(\tilde{\mathbf{x}}_i^k)$
  ▷ *Optimize objective $\alpha\mathcal{L}_2 + \mathcal{L}_{ML}$ wrt θ:*
  $\Delta\theta \leftarrow \nabla_\theta \frac{1}{N} \sum_i \alpha(E_\theta(\mathbf{x}_i^+)^2 + E_\theta(\mathbf{x}_i^-)^2) + E_\theta(\mathbf{x}_i^+) - E_\theta(\mathbf{x}_i^-)$
  Update θ based on $\Delta\theta$ using Adam optimizer
  $\mathcal{B} \leftarrow \mathcal{B} \cup \tilde{\mathbf{x}}_i$
end while

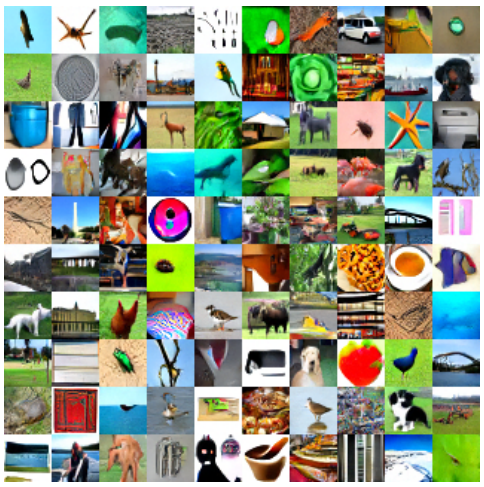


Figure 2: Conditional ImageNet32x32 EBM samples

4 Image Modeling

(a) GLOW Model (b) EBM (c) EBM (10 historical) (d) EBM Sample Buffer

Figure 3: Comparison of image generation techniques on unconditional CIFAR-10 dataset.

In this section, we show that EBMs are effective generative models for images. We show EBMs are able to generate high fidelity images and exhibit mode coverage on CIFAR-10 and ImageNet. We further show EBMs exhibit adversarial robustness and better out-of-distribution behavior than other likelihood models. Our model is based on the ResNet architecture (using conditional gains and biases per class [Dumoulin et al.] for conditional models) with details in the supplement. We present sensitivity analysis, likelihoods, and ablations in the supplement in A.4. We provide a comparison between EBMs and other likelihood models in A.5. Overall, we find that EBMs are both more parameter/computationally efficient than likelihood models, though worse than GANs.

4.1 Image Generation

We show unconditional CIFAR-10 images in Figure 3, with comparisons to GLOW [Kingma and Dhariwal, 2018], and conditional ImageNet32x32 images in Figure 2. We provide qualitative images of ImageNet128x128 and other visualizations in A.1.

Model	Inception*	FID
CIFAR-10 Unconditional		
PixelCNN [Van Oord et al., 2016]	4.60	65.93
PixelIQN [Ostrovski et al., 2018]	5.29	49.46
EBM (single)	6.02	40.58
DCGAN [Radford et al., 2016]	6.40	37.11
WGAN + GP [Gulrajani et al., 2017]	6.50	36.4
EBM (10 historical ensemble)	6.78	38.2
SNGAN [Miyato et al., 2018]	**8.22**	21.7
CIFAR-10 Conditional		
Improved GAN	8.09	-
EBM (single)	8.30	37.9
Spectral Normalization GAN	**8.59**	25.5
ImageNet 32x32 Conditional		
PixelCNN	8.33	33.27
PixelIQN	10.18	22.99
EBM (single)	**18.22**	**14.31**
ImageNet 128x128 Conditional		
ACGAN [Odena et al., 2017]	28.5	-
EBM* (single)	28.6	43.7
SNGAN	**36.8**	**27.62**

Figure 4: Table of Inception and FID scores for ImageNet32x32 and CIFAR-10. Quantitative numbers for ImageNet32x32 from [Ostrovski et al., 2018]. (*) We use Inception Score (from original OpenAI repo) to compare with legacy models, but strongly encourage future work to compare soley with FID score, since Langevin Dynamics converges to minima that artificially inflate Inception Score. (**) conditional EBM models for 128x128 are smaller than those in SNGAN.

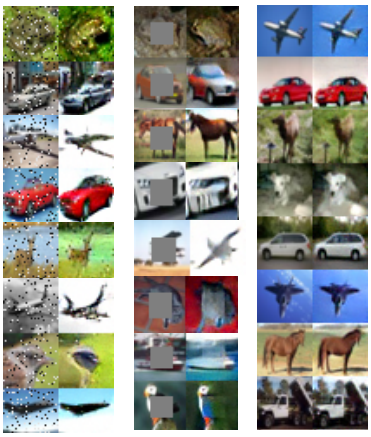


Figure 5: EBM image restoration on images in the **test** set via MCMC. The right column shows failure (approx. 10% objects change with ground truth initialization and 30% of objects change in salt/pepper corruption or inpainting. Bottom two rows shows worst case of change.)

We quantitatively evaluate image quality of EBMs with Inception score [Salimans et al., 2016] and FID score [Heusel et al., 2017] in Table 4. Overall we obtain significantly better scores than likelihood models PixelCNN and PixelIQN, but worse than SNGAN [Miyato et al., 2018]. We found that in the unconditional case, mode exploration with Langevin took a very long time, so we also experimented in *EBM (10 historical ensemble)* with sampling joint from the last 10 snapshots of the model. At training time, extensive exploration is ensured with the replay buffer (Figure 3d). Our models have similar number of parameters to SNGAN, but we believe that significantly more parameters may be necessary to generate high fidelity images with mode coverage. On ImageNet128x128, due to computational constraints, we train a smaller network than SNGAN and do not train to convergence.

4.2 Mode Evaluation

We evaluate over-fitting and mode coverage in EBMs. To test over-fitting, we plotted histogram of energies for CIFAR-10 train and test dataset in Figure 11 and note almost identical curves. In the supplement, we show that the nearest neighbor of generated images are not identical to images in the training dataset. To test mode coverage in EBMs, we investigate MCMC sampling on corrupted CIFAR-10 test images. Since Langevin dynamics is known to mix slowly [Neal, 2011] and reach local minima, we believe that good denoising after limited number of steps of sampling indicates probability modes at respective test images. Similarly, lack of movement from a ground truth test image initialization after the same number of steps likely indicates probability mode at the test image. In Figure 5, we find that if we initialize sampling with images from the test set, images do not move significantly. However, under the same number of steps, Figure 5 shows that we are able to reliably decorrupt masked and salt and pepper corrupted images, indicating good mode coverage. We note that large number of steps of sampling lead to more saturated images, which are due to sampling low temperature modes, which are saturated across likelihood models (see appendix). In comparison, GANs have been shown to miss many modes of data and cannot reliably reconstruct many different test images [Yeh et al.]. We note that such decorruption behavior is a nice property of implicit generation without need of explicit knowledge of corrupted pixels.

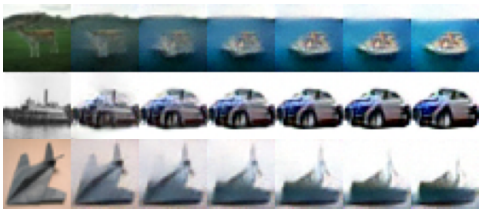


Figure 6: Illustration of cross-class implicit sampling on a conditional EBM. The EBM is conditioned on a particular class but is initialized with an image from a separate class.

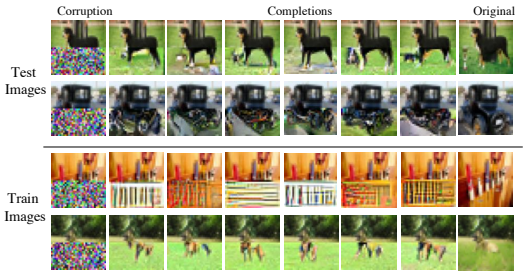


Figure 7: Illustration of image completions on conditional ImageNet model. Our models exhibit diversity in inpainting.

Another common test for mode coverage and overfitting is masked inpainting [Van Oord et al., 2016]. In Figure 7, we mask out the bottom half of ImageNet images and test the ability to sample the masked pixels, while fixing the value of unmasked pixels. Running Langevin dynamics on the images, we find diversity of completions on train/test images, indicating low overfitting on training set and diversity characterized by likelihood models. Furthermore initializing sampling of a class conditional EBM with images from images from another class, we can further test for presence of probability modes at images far away from the those seen in training. We find in Figure 6 that sampling on such images using an EBM is able to generate images of the target class, indicating semantically meaningful modes of probability even far away from the training distribution.

4.3 Adversarial Robustness

We show conditional EBMs exhibit adversarial robustness on CIFAR-10 classification, **without explicit** adversarial training. To compute logits for classification, we compute the negative energy of the image in each class. Our model, without fine-tuning, achieves an accuracy of 49.6%. Figure 8 shows adversarial robustness curves. We ran 20 steps of PGD as in [Madry et al., 2017], on the above logits. To undergo classification, we then ran 10 steps sampling initialized from the starting image (with a bounded deviation of 0.03) from each conditional model, and then classified using the lowest energy conditional class. We found that running PGD incorporating sampling was

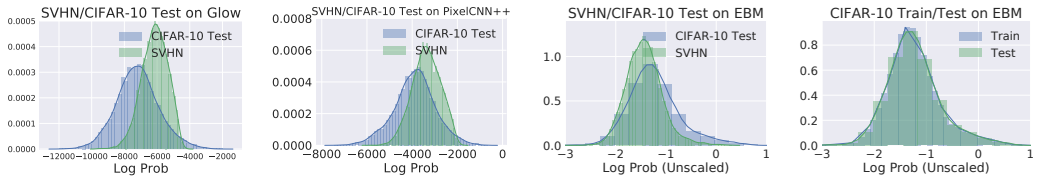
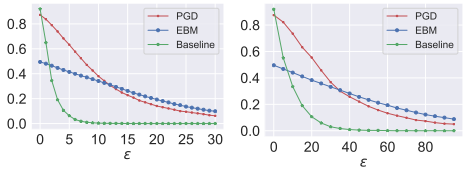


Figure 11: Histogram of relative likelihoods for various datasets for Glow, PixelCNN++ and EBM models

less successful than without. Overall we find in Figure 8 that EBMs are very robust to adversarial perturbations and outperforms the SOTA L_∞ model in [Madry et al., 2017] on L_∞ attacks with $\epsilon > 13$.



(a) L_∞ robustness (b) L_2 Robustness

Figure 8: ϵ plots under L_∞ and L_2 attacks of conditional EBMs as compared to PGD trained models in [Madry et al., 2017] and a baseline Wide ResNet18.

4.4 Out-of-Distribution Generalization

We show EBMs exhibit better out-of-distribution (OOD) detection than other likelihood models. Such a task requires models to have high likelihood on the data manifold and low likelihood at all other locations and can be viewed as a proxy of log likelihood. Surprisingly, Nalisnick et al. [2019] found likelihood models such as VAE, PixelCNN, and Glow models, are unable to distinguish data assign higher likelihood to many OOD images. We constructed our OOD metric following following [Hendrycks and Gimpel, 2016] using Area Under the ROC Curve (AUROC) scores computed based on classifying CIFAR-10 test images from other OOD images using relative log likelihoods. We use SVHN, Textures [Cimpoi et al., 2014], monochrome images, uniform noise and interpolations of separate CIFAR-10 images as OOD distributions. We provide examples of OOD images in Figure 9. We found that our proposed OOD metric correlated well with training progress in EBMs.

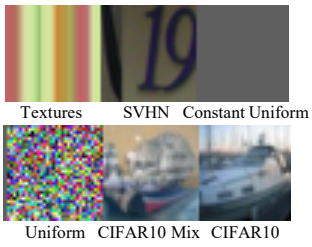


Figure 9: Illustration of images from each of the out of distribution dataset.

Model	PixelCNN++	Glow	EBM (ours)
SVHN	0.32	0.24	**0.63**
Textures	0.33	0.27	**0.48**
Constant Uniform	0.0	0.0	**0.30**
Uniform	1.0	1.0	**1.0**
CIFAR10 Interpolation	**0.71**	0.59	0.70
Average	0.47	0.42	**0.62**

Figure 10: AUROC scores of out of distribution classification on different datasets. Only our model gets better than chance classification.

In Table 10, unconditional EBMs perform **significantly better** out-of-distribution than other auto-regressive and flow generative models and have OOD scores of 0.62 while the closest, PixelCNN++, has a OOD score of 0.47. We provide histograms of relative likelihoods for SVHN in Figure 11 which are also discussed in [Nalisnick et al., 2019, Hendrycks et al., 2018]. We believe that the reason for better generalization is two-fold. First, we believe that the negative sampling procedure in EBMs helps eliminate spurious minima. Second, we believe EBMs have a flexible structure that allows global context when estimating probability without imposing constraints on latent variable structure. In contrast, auto-regressive models model likelihood sequentially, which makes global coherence difficult. In a different vein, flow based models must apply continuous transformations onto a continuous connected probability distribution which makes it very difficult to model disconnected modes, and thus assign spurious density to connections between modes.

5 Trajectory Modeling

We show that EBMs generate and generalize well in the different domain of trajectory modeling. We train EBMs to model dynamics of a simulated robot hand manipulating a free cube object [OpenAI, 2018]. We generated 200,000 different trajectories of length 100, from a trained policy (with every 4th action set to a random action for diversity), with a 90-10 train-test split. Models are trained to predict positions of all joints in the hand and orientation and position of the cube one step in the future. We test performance by evaluating many step roll-outs of self-predicted trajectories.

5.1 Training Setup and Metrics

We compare EBM models to feedforward models (FC), both of which are composed of 3 layers of 128 hidden units. We apply spectral normalization to FC to prevent multi-step explosion. We evaluate multi-step trajectories by computing Frechet Distance [Dowson and Landau, 1982] between predicted and ground distributions across all states at timestep t. We found this metric was a better metric of trajectories than multi-step MSE due to accumulation of error.

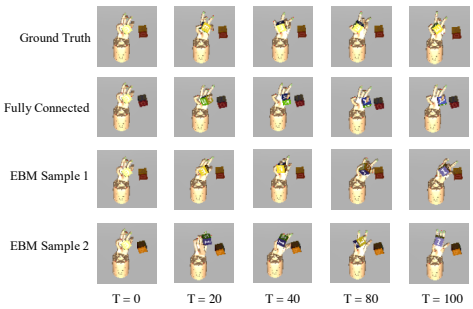


Figure 12: Views of hand manipulation trajectories generated unconditionally from the same state(1st frame).

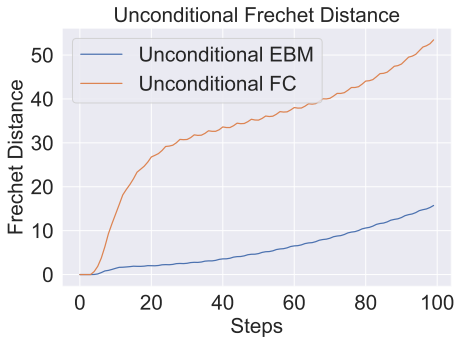


Figure 13: Conditional and Unconditional Modeling of Hand Manipulation through Frechet Distance

5.2 Multi-Step Trajectory Generation

We evaluated EBMs for both action conditional and unconditional prediction of multi-step rollouts. Quantitatively, by computing the average Frechet distance across all time-steps, unconditional EBM have value 5.96 while unconditional FC networks have a value of 33.28. Conditional EBM have value 8.97 while a conditional FC has value 19.75. We provide plots of Frechet distance over time in Figure 13. In Figure 13, we observe that for unconditional hand modeling in a FC network, the Frechet distance increases dramatically in the first several time steps. Qualitatively, we found that the same FC networks stop predicting hand movement after several several steps as demonstrated in Figure 12. In contrast, Frechet distance increases slowly for unconditional EBMs. The unconditional models are able to represent multimodal transitions such as different types of cube rotation and Figure 12 shows that the unconditional EBMs generate diverse realistic trajectories.

6 Online Learning

Method	Accuracy
EWC [Kirkpatrick et al., 2017]	19.80 (0.05)
SI [Zenke et al., 2017]	19.67 (0.09)
NAS [Schwarz et al., 2018]	19.52 (0.29)
LwF [Li and Snavely, 2018]	24.17 (0.33)
VAE	40.04 (1.31)
EBM (ours)	**64.99** (4.27)

Table 1: Comparison of various continual learning benchmarks. Values averaged acrossed 10 seeds reported as mean (standard deviation).

We find that EBMs also perform well in continual learning. We evaluate incremental class learning on the Split MNIST task proposed in [Farquhar and Gal, 2018]. The task evaluates overall MNIST digit classification accuracy given 5 sequential training tasks of disjoint pairs of digits. We train a conditional EBM with 2 layers of 400 hidden units work and compare with a generative conditional VAE baseline with both encoder/decoder having 2 layers of 400 hidden units. Additional training details are covered in the appendix. We train the generative models to represent the joint distribution of images and

labels and classify based off the lowest energy label. Hsu et al. [2018] analyzed common continual learning algorithms such as EWC [Kirkpatrick et al., 2017], SI [Zenke et al., 2017] and NAS [Schwarz et al., 2018] and find they obtain performance around 20%. LwF [Li and Snavely, 2018] performed the best with performance of 24.17 ± 0.33 , where all architectures use 2 layers of 400 hidden units. However, since each new task introduces two new MNIST digits, a test accuracy of around 20% indicates complete forgetting of previous tasks. In contrast, we found continual EBM training obtains **significantly higher** performance of 64.99 ± 4.27. All experiments were run with 10 seeds.

A crucial difference is that negative training in EBMs only locally "forgets" information corresponding to negative samples. Thus, when new classes are seen, negative samples are conditioned on the new class, and the EBM only forgets unlikely data from the new class. In contrast, the cross entropy objective used to train common continual learning algorithms down-weights the likelihood of all classes not seen. We can apply this insight on other generative models, by maximizing the likelihood of a class conditional model at train time and then using the highest likelihood class as classification results. We ran such a baseline using a VAE and obtained a performance of 40.04 ± 1.31, which is higher than other continual learning algorithms but less than that in a EBM.

7 Compositional Generation

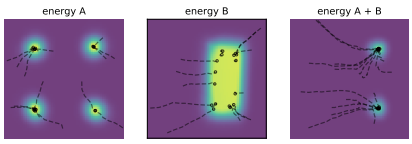


Figure 14: A 2D example of combining EBMs through summation and the resulting sampling trajectories.

Finally, we show compositionality through implicit generation in EBMs. Consider a set of conditional EBMs for separate independent latents. Sampling through the joint distribution on all latents is represented by generation on an EBM that is the sum of each conditional EBM [Hinton, 1999] and corresponds to a product of experts model. As seen in Figure 14, summation naturally allows composition of EBMs. We sample from joint conditional distribution through Langevin dynamics sequentially from each model.

We conduct our experiments on the dSprites dataset [Higgins et al., 2017], which consists of all possible images of an object (square, circle or heart) varied by scale, position, rotation with labeled latents. We trained conditional EBMs for each latent and found that scale, position and rotation worked well. The latent for shape was learned poorly, and we found that even our unconditional models were not able to reliably generate different shapes which was also found in [Higgins et al., 2017]. We show some results on CelebA in A.6.

Figure 15: Samples from joint distribution of 4 independent conditional EBMs on scale, position, rotation and shape (left panel) with associated ground truth rendering (right panel).

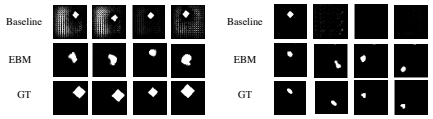


Figure 16: GT = Ground Truth. Images of cross product generalization of size-position (left panel) and shape-position (right panel).

Joint Conditioning In Figure 15, we provide generated images from joint conditional sampling. Under such sampling we are able to generate images very close to ground truth for all classes with exception of shape. This result also demonstrates mode coverage across all data.

Zero-Shot Cross Product Generalization We evaluate the ability of EBMs to generalize to novel combinations of latents. We generate three datasets, D1: different size squares at a central position, D2: smallest size square at each location, D3: different shapes at the center position. We evaluate size-position generalization by training independent energy functions on D1 and D2, and test on generating different size squares at all positions. We similarly evaluate shape-position generalization for D2 and D3. We generate samples at novel combinations by sampling from the summation of energy functions (we first finetune the summation energy to generate both training datasets using a KL term defined in the appendix). We compare against a joint conditional model baseline.

We present results of generalization in Figure 16. In the left panel of Figure 16, we find the EBMs are able to generalize to different sizes at different position (albeit with loss in sample quality) while a conditional model ignores the size latent and generates only images seen in the training. In the right

panel of Figure 16, we found that EBMs are able to generalize to combinations of shape and position by creating a distinctive shape for each conditioned shape latent at different positions (though the generated shape doesn't match the shape of the original shape latent), while a baseline is unable to generate samples. We believe the compositional nature of EBMs is crucial to generalize in this task.

8 Conclusion

We have presented a series of techniques to scale up EBM training. We further show unique benefits of implicit generation and EBMs and believe there are many further directions to explore. Algorithmically, we think it would be interesting to explore methods for faster sampling, such as adaptive HMC. Empirically, we think it would be interesting to explore, extend, and understand results we've found, in directions such as compositionality, out-of-distribution detection, adversarial robustness, and online learning. Furthermore, we think it may be interesting to apply EBMs on other domains, such as text and as a means for latent representation learning.

9 Acknowledgements

We would like to thank Ilya Sutskever, Alec Radford, Prafulla Dhariwal, Dan Hendrycks, Johannes Otterbach, Rewon Child and everyone at OpenAI for helpful discussions.

References

David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. A learning algorithm for boltzmann machines. *Cognit. Sci.*, 9(1):147–169, 1985.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Accurate and conservative estimates of mrf log-likelihood using reverse annealing. In *Artificial Intelligence and Statistics*, pages 102–110, 2015.

M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

Barry A Cipra. An introduction to the ising model. *The American Mathematical Monthly*, 94(10):937–959, 1987.

Peter Dayan, Geoffrey E Hinton, Radford M Neal, and Richard S Zemel. The helmholtz machine. *Neural Comput.*, 7(5):889–904, 1995.

Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

DC Dowson and BV Landau. The fréchet distance between multivariate normal distributions. *Journal of multivariate analysis*, 12(3):450–455, 1982.

Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style.

Sebastian Farquhar and Yarin Gal. Towards robust evaluations of continual learning. *arXiv preprint arXiv:1805.09733*, 2018.

Chelsea Finn, Paul Christiano, Pieter Abbeel, and Sergey Levine. A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models. In *NIPS Workshop*, 2016.

Ruiqi Gao, Yang Lu, Junpei Zhou, Song-Chun Zhu, and Ying Nian Wu. Learning generative convnets via multi-grid modeling and sampling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9155–9164, 2018.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. In *NIPS*, 2017.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. *arXiv preprint arXiv:1702.08165*, 2017.

Junxian He, Daniel Spokoyny, Graham Neubig, and Taylor Berg-Kirkpatrick. Lagging inference networks and posterior collapse in variational autoencoders. *arXiv preprint arXiv:1901.05534*, 2019.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *arXiv preprint*, 2018.

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 *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.

Irina Higgins, Loic Matthey, Arka Pal, Christopher P Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. Beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

Geoffrey Hinton, Simon Osindero, Max Welling, and Yee-Whye Teh. Unsupervised discovery of nonlinear structure using contrastive backpropagation. *Cognitive science*, 30(4):725–731, 2006.

Geoffrey E Hinton. Products of experts. *International Conference on Artificial Neural Networks*, 1999.

Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Training*, 14(8), 2006.

Yen-Chang Hsu, Yen-Cheng Liu, and Zsolt Kira. Re-evaluating continual learning scenarios: A categorization and case for strong baselines. *arXiv preprint arXiv:1810.12488*, 2018.

Taesup Kim and Yoshua Bengio. Deep directed generative models with energy-based probability estimation. *arXiv preprint arXiv:1606.03439*, 2016.

Diederik P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *arXiv preprint arXiv:1807.03039*, 2018.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in neural information processing systems*, pages 4743–4751, 2016.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Rithesh Kumar, Anirudh Goyal, Aaron Courville, and Yoshua Bengio. Maximum entropy generators for energy-based models. *arXiv preprint arXiv:1901.08508*, 2019.

Karol Kurach, Mario Lucic, Xiaohua Zhai, Marcin Michalski, and Sylvain Gelly. The gan landscape: Losses, architectures, regularization, and normalization. *arXiv preprint arXiv:1807.04720*, 2018.

Yann LeCun, Sumit Chopra, and Raia Hadsell. A tutorial on energy-based learning. 2006.

Zhengqi Li and Noah Snavely. Learning intrinsic image decomposition from watching the world. In *CVPR*, 2018.

Yang Liu, Prajit Ramachandran, Qiang Liu, and Jian Peng. Stein variational policy gradient. *arXiv preprint arXiv:1704.02399*, 2017.

Yang Lu, Song-Chun Zhu, and Ying Nian Wu. Learning frame models using cnn filters. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

Andriy Mnih and Geoffrey Hinton. *Learning nonlinear constraints with contrastive backpropagation*. Citeseer, 2004.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. In *NIPS Workshop*, 2013.

Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=H1xwNhCcYm`.

Radford M Neal. Annealed importance sampling. *Stat. Comput.*, 11(2):125–139, 2001.

Radford M Neal. Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2(11), 2011.

Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2642–2651. JMLR. org, 2017.

OpenAI. Learning dexterous in-hand manipulation. In *arXiv preprint arXiv:1808.00177*, 2018.

Georg Ostrovski, Will Dabney, and Rémi Munos. Autoregressive quantile networks for generative modeling. *arXiv preprint arXiv:1806.05575*, 2018.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016.

Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine learning*, 62(1-2):107–136, 2006.

Ruslan Salakhutdinov and Geoffrey E. Hinton. Deep boltzmann machines. In David A. Van Dyk and Max Welling, editors, *AISTATS*, volume 5 of *JMLR Proceedings*, pages 448–455. JMLR.org, 2009. URL `http://www.jmlr.org/proceedings/papers/v5/salakhutdinov09a.html`.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NIPS*, 2016.

Jonathan Schwarz, Jelena Luketina, Wojciech M Czarnecki, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. *arXiv preprint arXiv:1805.06370*, 2018.

Yee Whye Teh, Max Welling, Simon Osindero, and Geoffrey E Hinton. Energy-based models for sparse overcomplete representations. *Journal of Machine Learning Research*, 4(Dec):1235–1260, 2003.

Tijmen Tieleman. Training restricted boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th international conference on Machine learning*, pages 1064–1071. ACM, 2008.

Richard Turner. Cd notes. 2005.

Benigno Uria, Iain Murray, and Hugo Larochelle. Rnade: The real-valued neural autoregressive density-estimator. In *Advances in Neural Information Processing Systems*, pages 2175–2183, 2013.

Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *ICML*, 2016.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 681–688, 2011.

Ying Nian Wu, Song Chun Zhu, and Xiuwen Liu. Equivalence of julesz ensembles and frame models. *International Journal of Computer Vision*, 38(3):247–265, 2000.

Yuhuai Wu, Yuri Burda, Ruslan Salakhutdinov, and Roger Grosse. On the quantitative analysis of decoder-based generative models. *arXiv preprint arXiv:1611.04273*, 2016.

Jianwen Xie, Wenze Hu, Song-Chun Zhu, and Ying Nian Wu. Learning sparse frame models for natural image patterns. *International Journal of Computer Vision*, 114(2-3):91–112, 2015.

Jianwen Xie, Yang Lu, Ruiqi Gao, Song-Chun Zhu, and Ying Nian Wu. Cooperative training of descriptor and generator networks. *arXiv preprint arXiv:1609.09408*, 2016a.

Jianwen Xie, Yang Lu, Song-Chun Zhu, and Ying Nian Wu. Inducing wavelets into random fields via generative boosting. *Applied and Computational Harmonic Analysis*, 41(1):4–25, 2016b.

Jianwen Xie, Yang Lu, Song-Chun Zhu, and Yingnian Wu. A theory of generative convnet. In *International Conference on Machine Learning*, pages 2635–2644, 2016c.

Jianwen Xie, Yang Lu, Ruiqi Gao, and Ying Nian Wu. Cooperative learning of energy-based model and latent variable model via mcmc teaching. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Raymond A Yeh, Chen Chen, Teck-Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3987–3995. JMLR. org, 2017.

Hongyi Zhang, Yann N. Dauphin, and Tengyu Ma. Residual learning without normalization via better initialization. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=H1gsz30cKX`.

Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016.

Song Chun Zhu, Ying Nian Wu, and David Mumford. Filters, random fields and maximum entropy (FRAME): towards a unified theory for texture modeling. *IJCV*, 27(2):107–126, 1998. doi: 10.1023/A:1007925832420.

A Appendix

A.1 Additional Qualitative Evaluation

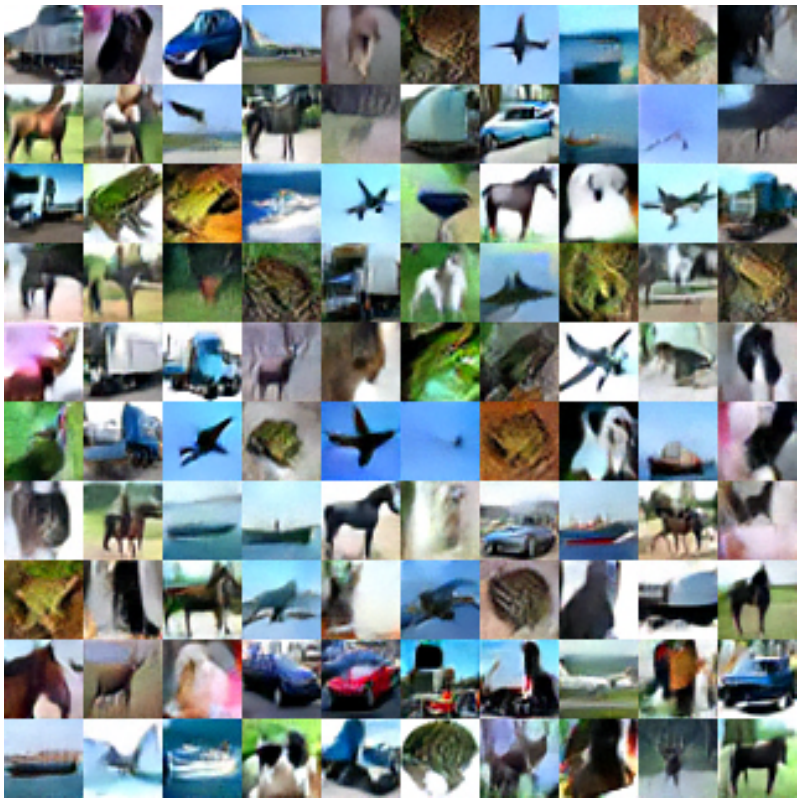


Figure 17: MCMC samples from conditional CIFAR-10 energy function

We present qualitative images from conditional generation on CIFAR10 in Figure 17 and from conditional generation of ImageNet128x128 in Figure 18.

We provide further images of cross class conversions using a conditional EBM model in Figure 19. Our model is able to convert images from different classes into reasonable looking images of the target class while sometimes preserving attributes of the original class.

We analyze nearest neighbors of images we generate in L2 distance Figure 20 and in Resnet-50 embedding space in Figure 21.

A.2 Test Time Sampling Process

We provide illustration of image generation from conditional and unconditional EBM models starting from random noise in Figure 22 with small amounts of random noise added. Dependent on the image generated there is slight drift from some start image to a final generated image. We typically observe that as sampling continues, much of the background is lost and a single central object remains.

We find that if small amounts of random noise are added, all sampling procedures generate a large initial set of diverse, reduced sample quality images before converging into a small set of high probability/quality image modes that are modes of images in CIFAR10. However, we find that if sufficient noise is added during sampling, we are able to slowly cycle between different images with larger diversity between images (indicating successful distribution sampling) but with reduced sample quality.

Due to this tradeoff, we use a replay buffer to sample images at test time, with slightly high noise then used during training time. For conditional energy models, to increase sample diversity, during initial image generation, we flip labels of images early on in sampling.

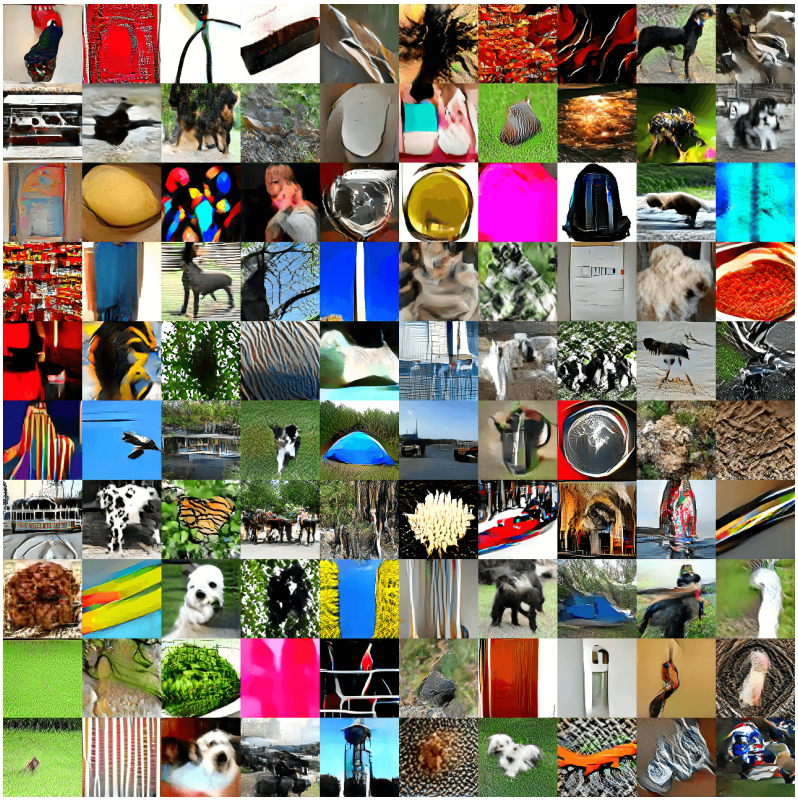


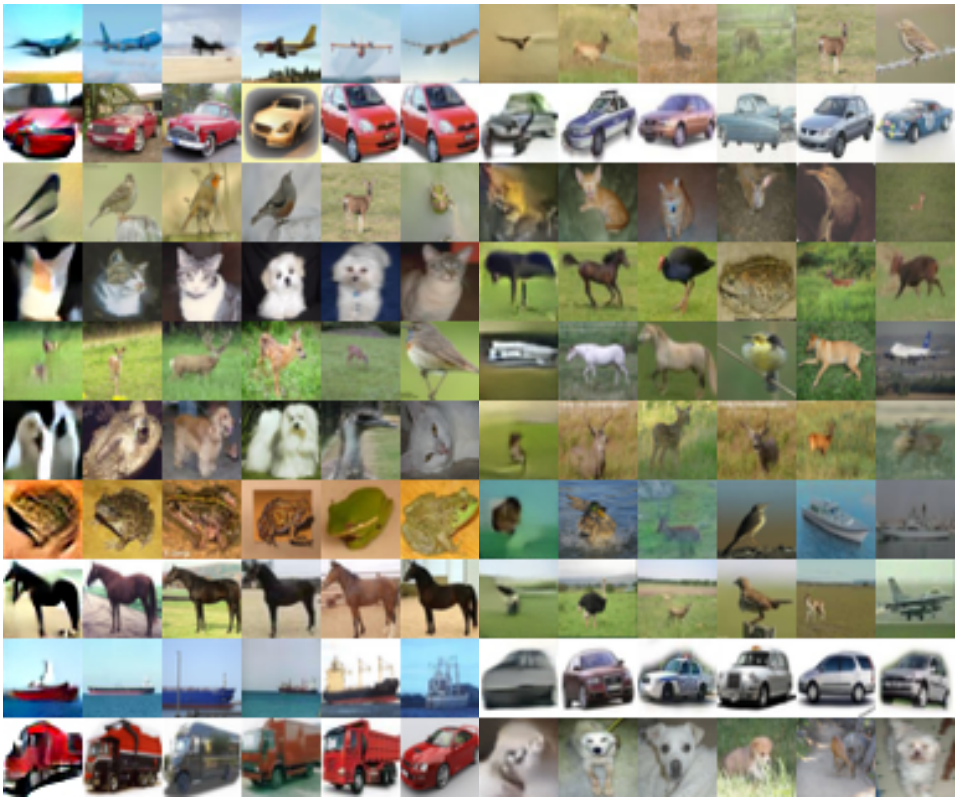
Figure 18: MCMC samples from conditional ImageNet128x128 models

Figure 19: Illustration of more cross class conversion applying MCMC on a conditional EBM. We condition on a particular class but is initialized with an image from a another class(left). We are able to preserve certain aspects of the image while altering others

A.3 Likelihood Evaluation And Ablations

To evaluate the likelihood of EBMs, we use AIS [Neal, 2001] and RAISE to obtain a lower bound of partition function [Burda et al., 2015]. We found that our energy landspaces were smooth and gave sensible likelihood estimates across a range of temperatures and so chose the appropriate temperature that maximized the likelihood of the model. When using these methods to estimate the partition function on CIFAR-10 or ImageNet, we found that it was too slow to get any meaningfull partition function estimates. Specifically, we ran AIS for over 300,000 chains (which took over 2 days of time) and still a very large gap between lower and upper partition function estimates.

While it was difficult to apply on CIFAR-10, we were able to get lower differences between upper and lower partition functions estimates on continuous MNIST. We rescaled MNIST and to be between



(a) Nearest neighbor images in CIFAR10 for condi-(b) Nearest neighbor images in CIFAR10 for unconditional energy models (leftmost generated, seperate class per row).

conditional energy model (leftmost generated)

Figure 20: Nearest neighbor images $L2$ distance for images generated from implicit sampling.

0 and 1 and added 1/256 random noise following [Uria et al., 2013]. Table 23 provides a table of log likelihoods on continuous MNIST across Flow, GAN, and VAE models as well as well as a comparison towards using PCD as opposed to a replay buffer to train on continuous MNIST. We find that the replay buffer is essential to good generation and likelihood, with the ablation of training with PCD instead of replay buffer getting significantly worse likelihood. We further find that EBMs appear to compare favorably to other likelihood models.

A.4 Hyper-parameter Sensitivity

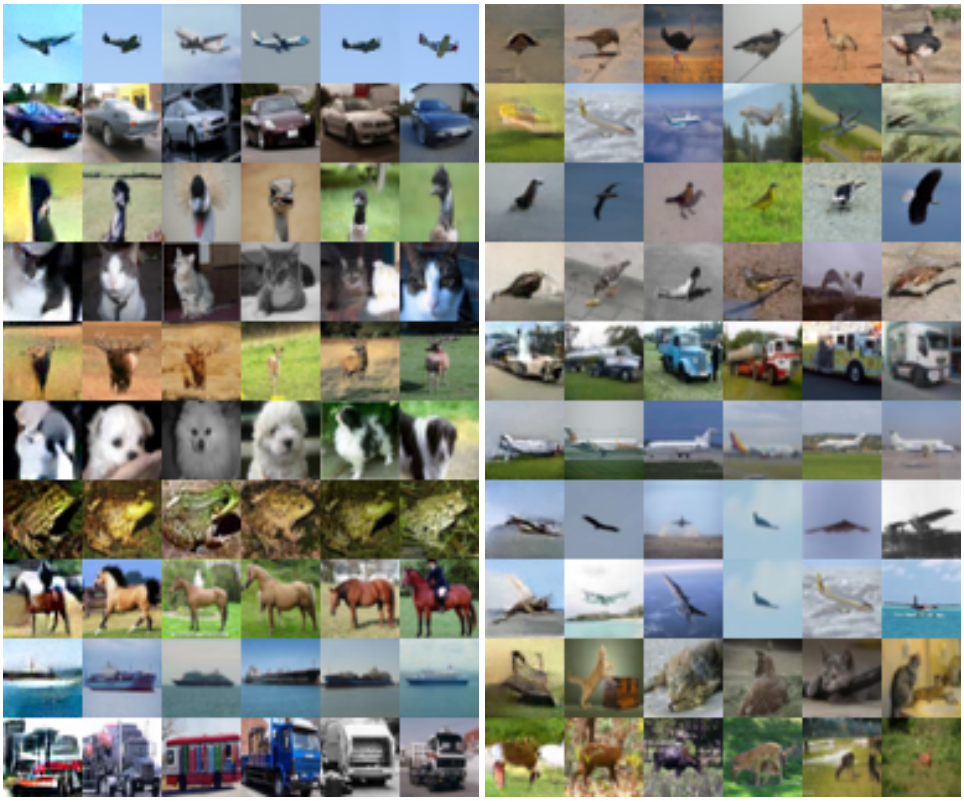
Empirically, we found that EBM training under our technique was relatively insensitive to the hyper-parameters. For example, Table 25 shows log likelihoods on continuous MNIST across several different order of magnitudes of L2 regularization and step size magnitude. We find consistent likelihood and good qualitative generation across different variations of L2 coefficient and step size magnitude and observed similar results in CIFAR-10 and Imagenet. Training is insensitive to replay buffer size (as long as size is greater than around 10000 samples).

Models	Parameters	Training Time	Sampling Time
EBM	5M	48	3 Hour (Variable)
PixelCNN++	160M	1300	72 Hour
Glow	115M	1300	0.5 Hour
SNGAN	5M	9	0.02 Hour

Figure 24: Comparison of parameters, training time (GPU hours), and sampling time (for 50000 images) on CIFAR-10. For EBM, sampling time depends on steps of sampling. We used 3 hours of sampling to generate quantitative metrics, but sampling can be much faster (around 0.2 hour) with reduced diversity.

A.5 Comparison With Other Likelihood Models

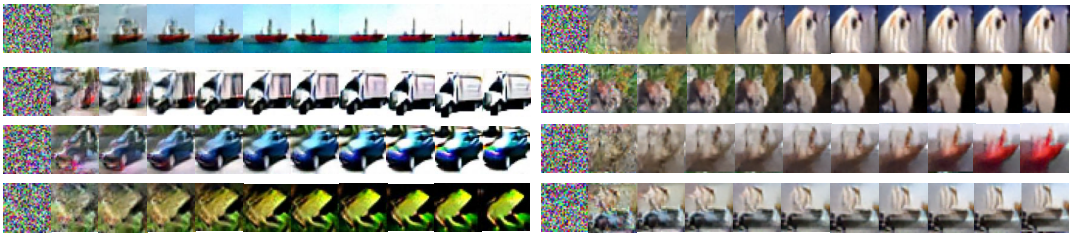
We compare EBMs to other generative models in Figure 24 on CIFAR-10. EBMs are faster to train than other likelihood models, with fewer parameters, but are more expensive than GAN based models (due to Langevin dynamics sampling), and slower to sample. Training time for PixelCNN++ and



(a) Nearest neighbor images in CIFAR10 for conditional energy models (leftmost generated, seperate class per row).

(b) Nearest neighbor images in CIFAR10 for unconditional energy model (leftmost generated)

Figure 21: Nearest neighbor images ResNet-50 distance for images generated from implicit sampling.



(a) Illustration of implicit sampling on conditional EBM of CIFAR-10

(b) Illustration of implicit sampling on an unconditional model on CIFAR-10

Figure 22: Generation of images from random noise.

Glow are from reported values in their papers, while sampling time and parameters were obtained from released code repositories. We have added the table to the appendix of the paper and added discussion on these trade-offs and intractability of likelihood evaluation in the main paper.

A.6 Image Saturation

When EBMs are run for a large number of sampling steps, images appear in increased saturation. This phenomenon can be exampled by the fact that many steps of sampling typically converge to high likelihood modes. Somewhat unintuitively, as seen also by out of distribution performance of likelihood models, such high likelihood modes on likelihood models trained on real datasets are often very texture based and heavily saturated. We provide illustration of this phenomenon on Glow in Figure 26.

Model	Lower Bound	Upper Bound
EBM + PCD	380.9	482
GAN 50 [Wu et al., 2016]	618.4	636.1
VAE 50 [Wu et al., 2016]	985.0	997.1
NICE [Dinh et al., 2014]	**1980.0**	1980.0
EBM + Replay Buffer	1925.0	**2218.3**

Figure 23: Log likelihood in Nats on Continuous MNIST. EBMs are evaluated by running AIS for 10000 chains

Hyper-parameter	Value	Lower Bound	Upper Bound
L2 Coefficient	0.01	1519	2370
	0.1	1925	2218
	1.0	1498	2044
Step Size	10.0	1498	2044
	100.0	1765	2309
	1000.0	1740	2009

Figure 25: Log likelihood in Nats on Continuous MNIST under different settings of the L2 penalty coefficient and Langevin Step Size evaluated after running AIS and RAISE for 10000 chains. Lower and upper bound in likelihood remain relatively constant across several different order of magnitude of variation

Figure 26: Low Temperature (High likelihood mode) vs High Temperature (Low Likelihood mode) in Glow Model

A.7 Details on Continual Learning Training

To train EBM models on the continual learning scenario of Split MNIST, we train an EBM following Algorithm 1 in the main body of the paper. Initially, negative sampling is done with labels of the digits 0 and 1. Afterwards, negative sampling is done with labels of the digits 2 and 3 and so forth. Simultaneously, we train EBMs on ground truth image label annotations. We maintain a replay buffer of negative samples to enable effective training of the EBM.

A.8 KL Term

In cases of very highly peaked data, we can further regularize E such that q matches p by minimizing KL divergence between the two distributions:

$$\mathcal{L}_{\text{KL}}(\theta) = \text{KL} q_\theta p = \mathbb{E}_{\tilde{\mathbf{x}} \sim q_\theta} \left[\bar{E}(\tilde{\mathbf{x}}) \right] + \mathcal{H}\left[q_\theta\right] \tag{3}$$

Where $\bar{E}$ is treated as a constant target function that does not depend on θ. Optimizing the above loss requires differentiating through the Langevin dynamics sampling procedure of (1), which is possible since the procedure is differentiable. Intuitively, we train energy function such that a limited number of gradient-based sampling steps takes samples to regions of low energy. We only use the above term when fine-tuning combinations of energy functions in zero shot combination and thus ignore the entropy term.

The computation of the entropy term $\mathcal{H}\left[q_\theta\right]$ can resolved by approaches [Liu et al., 2017] propose an optimization procedure where this term is minimized by construction, but rely on a kernel function $\kappa(\mathbf{x}, \mathbf{x}')$, which requires domain-specific design. Otherwise, the entropy can also be resolved by adding a IAF [Kingma et al., 2016] to map to underlying Gaussian through which entropy can be evaluated.

Figure 27: Illustration of test time generation of various combinations of two independently trained EBMs conditioned on latents on gender, hair color, attractiveness, and age on CelebA.

3x3 conv2d, 128
ResBlock down 128
ResBlock 128
ResBlock down 256
ResBlock 256
ResBlock down 256
ResBlock 256
Global Sum Pooling
dense $\rightarrow$ 1

(a) Conditional CIFAR-10 Model

3x3 conv2d, 128
ResBlock down 128
ResBlock 128
ResBlock 128
ResBlock down 256
ResBlock 256
ResBlock 256
ResBlock down 256
ResBlock 256
ResBlock 256
Global Sum Pooling
dense $\rightarrow$ 1

(b) Unconditional CIFAR-10 Model

3x3 conv2d, 128
ResBlock down 256
ResBlock 256
ResBlock down 512
ResBlock 512
ResBlock down 1024
ResBlock 1024
Global Sum Pooling
dense $\rightarrow$ 1

(c) Conditional ImageNet32x32 Model

3x3 conv2d, 64
ResBlock down 64
ResBlock down 128
ResBlock down 256
ResBlock down 512
ResBlock down 1024
ResBlock 1024
Global Sum Pooling
dense $\rightarrow$ 1

(d) Conditional ImageNet128x128 Model

A.9 Additional Compositionality Results

We show additional compositionality results on the CelebA dataset. We trained seperate conditional EBMs on the latents attractiveness, hair color, age, and gender. We show different combinations of two conditional models in Figure 27.

A.10 Model

We use the residual model in Figure 28a for conditional CIFAR-10 images generation and the residual model in Figure 28b for unconditional CIFAR10 and Imagenet images. We found unconditional models need additional capacity. Our conditional and unconditional architectures are similar to architectures in [Miyato et al., 2018].

We found definite gains with additional residual blocks and wider number of filters per block. Following [Zhang et al., 2019, Kingma and Dhariwal, 2018], we initialize the second convolution of residual block to zero and a scalar multiplier and bias at each layer. We apply spectral normalization on all weights. When using spectral normalization, zero weight initialized convolution filters were instead initialized from random normals with standard deviations of 1^{-10} (with spectrum normalized to be below 1). We use conditional bias and gains in each residual layer for a conditional model. We found it important when down-sampling to do average pooling as opposed to strided convolutions. We use leaky ReLUs throughout the architecture.

We use the architecture in Figure 18 for generation of conditional ImageNet32x32 images.

A.11 Training Details and Hyperparameters

For CIFAR-10 experiments, we use 60 steps of Langevin dynamics to generate negative samples. We use a replay buffer of size of 10000 image. We scale images to be between 0 and 1. We clip gradients to have individual value magnitude of less than 0.01 and use a step size of 10 for each gradient step of

Langevin dynamics. The L2 loss coefficient is set to 1. We use random noise with standard deviation $\lambda = 0.005$. CIFAR-10 models are trained on 1 GPU for 2 days. We use the Adam Optimizer with $\beta_1 = 0.0$ and $\beta_2 = 0.999$ with a training learning rate of 10^{-4}. We use a batch size during training of 128 positive and negative samples. For both experiments, we clip all training gradients that are more than 3 standard deviations from the 2nd order Adam parameters. We use spectral normalization on networks. For ImageNet32x32 images, we an analogous setup with models are trained for 5 days using 32 GPUs. For ImageNet 128x128, we use a step size 100 and train for 7 days using 32 GPUs.

For robotic simulation experiments we used 10 steps of Langevin dynamics to generate negative samples, but otherwise use identical settings as for image experiments.

A.12 Tips And Failures

We provide a list of tips, observations and failures that we observe when trying to train energy based models. We found evidence that suggest the following observations, though in no way are we certain that these observations are correct.

We found the following tips useful for training.

- We found that EBM training is most sensitive to MCMC transition step sizes (though there is around 2 to 3 order of magnitude that MCMC transition steps can vary).
- We found that that using either ReLU, LeakyReLU, or Swish activation in EBMs lead to good performance. The Swish activation in particular adds a noticeable boost to training stability.
- When using residual networks, we found that performance can be improved by using 2D average pooling as opposed to transposed convolutions
- We found that group, layer, batch, pixel or other types of normalization appeared to significantly hurt sampling, likely due to making MCMC steps dependent on surrounding data points.
- During a typical training run, we keep training until the sampler is unable to generate effective samples (when energies of proposal samples are much larger than energies of data points from the training data-set). Therefore, to extend training, the number of sampling steps to generate a negative sample can be increased.
- We find a direct relationship between depth / width and sample quality. More model depth or width can easily increase generation quality.
- When tuning noise when using Langevin dynamics, we found that very low levels of noise led to poor results. High levels of noise allowed large amounts of mode exploration initially but quickly led to early collapse of training due to failure of the sampler (failure to explore modes). We recommend keeping noise fixed at 0.005 and tune the step size per problem (though we found step sizes of around 10-100 work well).

We also tried the approaches below with the relatively little success.

- We found that training ensembles of energy functions (sampling and evaluating on ensembles) to help a bit, but was not worth the added complexity.
- We found it difficult to apply vanilla HMC to EBM training as optimal step sizes and leapfrog simulation numbers differed greatly during training, though applying adaptive HMC would be an interesting extension.
- We didn't find much success with adding a gradient penalty term as it seems to hurt model capacity.
- We tried training a separate network to help parameterize MCMC sampling but found that this made training unstable. However, we did find that using some part of the original model to parameterize MCMC (such as using the magnitude to energy to control step size) to help performance.

A.13 Relative Energy Visualization

In Figure 29, we show the energy distribution from $q(x)$ and from $p_d(x)$. We see that both distributions match each other relatively closely, providing evidence that $q(x)$ is close to $p(x)$

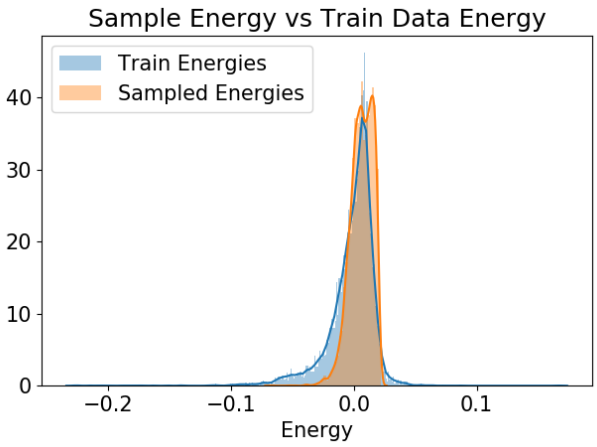


Figure 29: Relative energy of points sampled from $q(x)$ compared to CIFAR-10 train data points. We find that $q(x)$ exhibits a similar distribution to $p_d(x)$ and thus is similar to $p(x)$.