Implicit Generation and Generalization in Energy-Based Models

Yilun Du [1 2] **Igor Mordatch** [2]

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 show its success on the high-dimensional data domains of ImageNet32x32, ImageNet128x128, CIFAR-10, and robotic hand trajectories, achieving significantly better samples than other likelihood models and on par with contemporary GAN approaches, while covering all modes of the data. We highlight unique capabilities of implicit generation, such as energy compositionality and corrupt image reconstruction and inpainting. Finally, we show that EBMs generalize well and are able to achieve state-of-the-art out-of-distribution classification, exhibit adversarially robust classification, coherent long term predicted trajectory roll-outs, and generate zero-shot compositions of models. *

1 Introduction

In this work, we advocate for using continuous energy-based models (EBMs), represented as neural networks, for generative tasks and as a means for generalizable models. 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. They also 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 has a number of conceptual advantages compared to methods that use explicit functions to generate samples, such as Variational Autoencoders (VAEs) (Kingma and Welling, 2014) and Generative Adversarial Networks (GANs) (Goodfellow et al., 2014).

Simplicity and Stability: An energy network is the only object that needs to be trained and designed in the model. There is no need to tune training processes for separate networks to make sure they are balanced (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)). There is also no need to design two separate network architectures and ensure their architectures are compatible and balanced.

Sharing of Statistical Strength: Since energy network 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, latent representations, or recurrent memory dynamics as opposed to developing them independently and redundantly in separate networks. This moves us towards learning architectures consisting of a single monolithic object, rather than a collection of independent parts.

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. This allows for a system that can make fast coarse guesses, make more deliberate inferences by running the optimization process longer. It also allows a system to refine external guesses by initializing optimization process with them.

Flexibility Of Generation: The power of 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 learn to model particular regions as high or lower energy.

Adaptive Generation: While the final objective of training an EBM looks similar to that of GANs, the generator is implicitly defined by the probability distribution, and automatically adapts as the distribution changes. As a result, the generator does not need to be trained, allowing EBMs to applied to domains where it is difficult to train the generator of a GAN as well as ameliorating mode collapse.

Compositionality: If we think of energy functions as costs for a certain goals or constraints, summation of two or more

[1]MIT, Boston, USA [2]OpenAI, San Francisco, USA. Correspondence to: Yilun Du <yilundu@mit.edu>.

*Additional results, source code, and pre-trained models are available at https://sites.google.com/view/igebm

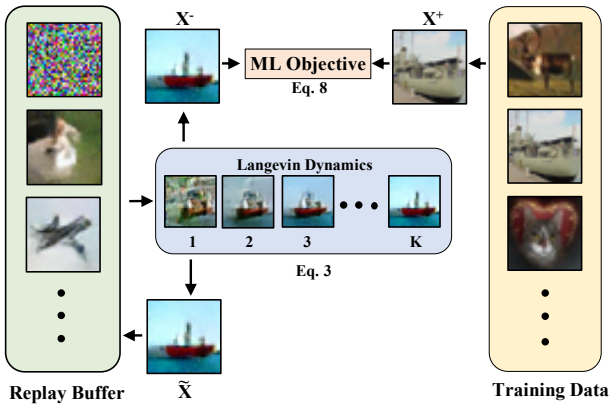


Figure 1: Overview of our method and the interrelationship of the multiple components involved.

energies corresponds to satisfying all their goals or constraints (Mnih and Hinton, 2004; Haarnoja et al., 2017). While such composition is simply a summation of 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 (and cannot be represented as a combination of constituent generators). This quality could allow energy-based models to more easily learn expressions for combinations of concepts.

Despite these advantages, energy-based models with implicit generation have been difficult to use on complex high-dimensional data domains. Implicit generation processes have primarily relied on gradient-free MCMC methods such as random walk or Gibbs sampling initialized from the data distribution which suffer from long mixing times. In this work, we overcome this issue by using Langevin dynamics (Welling and Teh, 2011), which uses gradient information for effective sampling and initialize 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 process to allow mixing between chains. To ensure effective sampling throughout the duration of training, we apply spectral normalization to smooth the sampling landscape and L2 normalization to constrain maximum energies. An overview of our approach is presented in Figure 1.

Empirically, we show that energy-based models trained on CIFAR-10 or ImageNet image datasets generate higher quality image samples than autoregressive or flow-based models and are on par with contemporary GANs approaches, while not suffering from mode collapse. The models exhibit generalization properties such as correctly assigning lower likelihood to out of distribution images than other methods (no spurious modes), generating a diversity of plausible image completions (covers all data modes) and being as resistant to adversarial perturbations as methods explicitly trained for these attacks. Our model also provides a number of unique capabilities, such as ability to 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 built in compositionality, and corrupt image reconstruction and inpainting. Finally, we show that energy-based models generalize well, on tasks such as out of domain generalization, adversarial robust classification, and multi-step trajectory prediction. With this work we hope to motivate more consideration and adoption of energy-based models in the future.

2 Related Work

Energy-based models (EBMs) have a long history in machine learning. (Dayan et al., 1995; 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 sample the partition function through amortized generation. (Kim and Bengio, 2016; Zhao et al., 2016; Haarnoja et al., 2017) propose learning a separate network to generate samples, which makes these methods closely connected to generative adversarial networks as shown by (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 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), propose 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. To reduce the time needed to generate samples, we use gradient based MCMC

(Langevin Dynamics) for more efficient sampling. 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. 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 can be used to define a probability distribution via the Boltzmann distribution, where $Z(\theta)$ denotes the partition function:

$$p_\theta(\mathbf{x}) = \frac{\exp(-E_\theta(\mathbf{x}))}{Z(\theta)} \tag{1}$$

$$Z(\theta) = \int \exp(-E_\theta(\mathbf{x}))d\mathbf{x} \tag{2}$$

Generating samples from this distribution is challenging, with previous work relying on MCMC methods such as random walk or Gibbs sampling (Hinton, 2006) which has long mixing times, especially for high-dimensional complex data such as images. To improve the mixing time of the sampling procedure, we propose the use of Langevin dynamics which makes use of the gradient of the energy function

$$\tilde{\mathbf{x}}^k = \tilde{\mathbf{x}}^{k-1} - \frac{\lambda}{2}\nabla_{\mathbf{x}} E_\theta(\tilde{\mathbf{x}}^{k-1}) + \omega^k,\ \omega^k \sim \mathcal{N}(0, \lambda) \tag{3}$$

$$\tilde{\mathbf{x}}^K \sim q_\theta \tag{4}$$

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 $q_\theta \to p$ and this procedure generates samples 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 (3) 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 via (1) and secondly it defines an implicit data generator via (3).

3.1 Maximum Likelihood Training

We want the distribution defined by E to model the data distribution p_D, which we do by minimizing negative log likelihood of the data

$$\mathcal{L}_{\text{ML}}(\theta) = \mathbb{E}_{\mathbf{x}\sim p_D}\left[-\log p_\theta(\mathbf{x})\right], \text{where} \tag{5}$$

$$-\log p_\theta(\mathbf{x}) = E_\theta(\mathbf{x}) - \log Z(\theta) \tag{6}$$

This objective is known to have the gradient (see (Turner, 2005) for derivation) $\nabla_\theta \mathcal{L}_{\text{ML}} =$

$$\mathbb{E}_{\mathbf{x}^+\sim p_D}\left[\nabla_\theta E_\theta(\mathbf{x}^+)\right] - \mathbb{E}_{\mathbf{x}^-\sim p_\theta}\left[\nabla_\theta E_\theta(\mathbf{x}^-)\right] \tag{7}$$

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 trained so far. In general, it is not tractable to generate samples from p_θ for the second term of above equation, and we rely on Langevin dynamics process in (3) to approximate this distribution: $\nabla_\theta \mathcal{L}_{\text{ML}} \approx$

$$\mathbb{E}_{\mathbf{x}^+\sim p_D}\left[\nabla_\theta E_\theta(\mathbf{x}^+)\right] - \mathbb{E}_{\mathbf{x}^-\sim q_\theta}\left[\nabla_\theta E_\theta(\mathbf{x}^-)\right] \tag{8}$$

Note that this is similar to the gradient of the Wasserstein GAN objective (Arjovsky et al., 2017), except with an implicit MCMC generating procedure, and no gradient through the sampling distribution.

The approximation in (8) is exact when Langevin dynamics procedure in (3) generates samples from p, which happens as $K \to \infty$ and $\lambda \to 0$. We find that in practice, p_d and q tend to appear to match each other in energy distribution as seen in Figure 2, showing that in our training it is likely that p matches q. Since our generator is implicitly defined by the original energy function, there is no need for an objective to train the generator, unlike in GANs. We note that even in cases when a particular chain does not mix, since our initial proposal distribution is a uniform distribution, all modes should be equally likely to be explored.

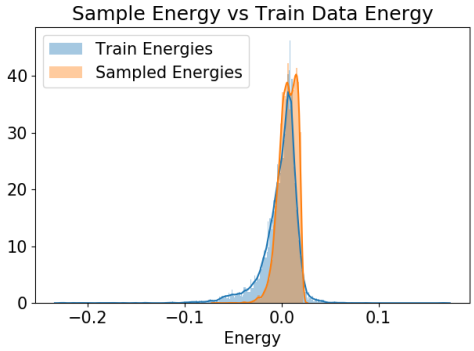


Figure 2: Relative energy of points sampled from $q(x)$ compared to CIFAR-10 train data points. We find that $q(x)$ exhibits a relatively similar distribution to $p_d(x)$.

3.2 Sample Replay Buffer

Langevin dynamics does not place restrictions on the sample initialization $\tilde{\mathbf{x}}^0$, but it plays an important role in the quality of samples in the truncated case and affects mode exploration. Persistent Contrastive Divergence (PCD) (Tieleman,

[†]Deterministic case of procedure in (3) is $\mathbf{x} = \arg\min E(\mathbf{x})$, which makes connection to implicit functions more clear.

2008) attempted to maintain 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 past samples or uniform noise to initialize Langevin dynamics procedure. This has the benefit of continuing to refine past samples and effectively increases number of sampling steps K (similar to PCD) and to improve 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 find that constraining the Lipschitz constant of the energy network can ameliorate these issues and allow most architectures blocks (such as residual and self-attention blocks) to be used in the model. 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 regularize L2 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 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.

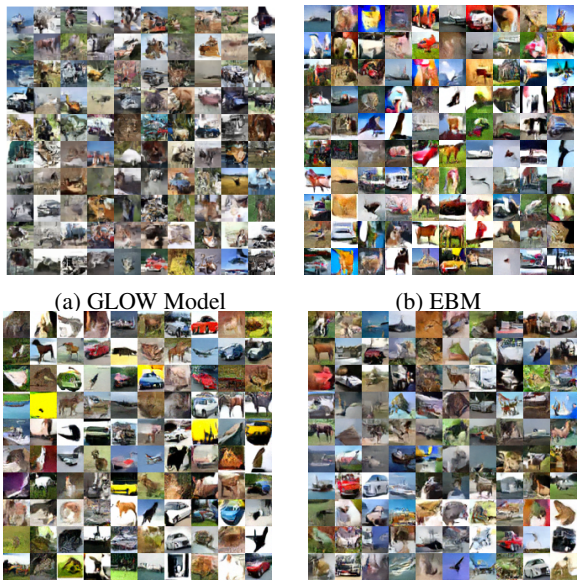
For completeness, we present the algorithm below, where $\Omega(\cdot)$ indicates the stop gradient operator..

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

4 Image Modeling

In this section, we show that EBMs are effective generative models of images in CIFAR-10 and ImageNet32x32 datasets - being able to generate high-quality images while simultaneously exhibiting modes at all data.We find that EBMs generate sharp images similar to contemporary GAN methods, while maintaining similar likelihood between train and test sets, and exhibit good out-of-distribution behavior and adversarial robustness.



(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.

Our model is based on the ResNet architecture (using conditional gains and biases per class (Dumoulin et al.)) with architecture and hyperparameters in the Appendix. Our models comparatively smaller, with often less than 10% the parameters of other models. We show quantitative numbers that our approach is stable to significant variation of hyper-parameters in A.4, present likelihoods and ablation in A.3.

4.1 Image Generation

We first evaluate the ability of EBMs to generate sharp samples on unconditional and conditional CIFAR-10 and ImageNet 32x32 datasets, which are comparatively better than other likelihood based models. To qualitatively evaluate image quality, we provide unconditional CIFAR-10 images in Figure 3 and conditional ImageNet32x32 images in Figure 4. In Figure 3, we see that compared to state of the art likelihood model, GLOW (Kingma and Dhariwal, 2018), our model is able to make more coherent artifact-free images. We further show that approach scales to larger data-sets, such as the ImageNet128x128 data-set. Examples of these and other samples along with visualizations of the sampling process can be found in the A.1.

We quantitatively evaluate image quality of EBMs with Inception score (Salimans et al., 2016) and FID score (Heusel

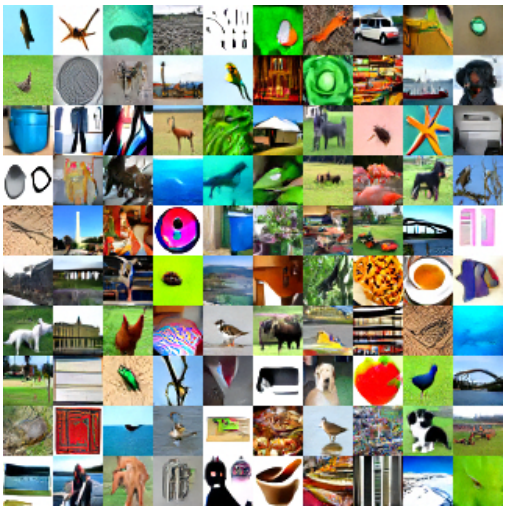


Figure 4: Conditional ImageNet32x32 EBM samples

et al., 2017) in Table 5. On unconditional CIFAR-10, we find higher inception scores and lower FID scores than PixelCNN and followup work such as PixelIQN (Ostrovski et al., 2018). We found that Langevin dynamics sampling explored limited modes, a problem mitigated by a replay buffer during training time. To mimic a replay buffer, in *EBM (10 historical ensemble)* condition, we sample jointly from the last 10 snapshots of the EBM.Our numbers are lower than those in SNGAN (Miyato et al., 2018) with similar parameters. We believe this is partly due to more capacity needed to model likelihood at all images and limited exploration in the time allotted. Qualitatively, we can see in Figure 3d that training exhibits many more modes than sampling using the last 10 snapshots (Figure 3c).

For conditional generation, we find that our inception scores are close to those of SNGAN on CIFAR-10. We believe a large reason for increase in performance of conditional EBMs relative to unconditional EBMs is improved mode exploration during evaluation / training time. With conditional EBMs, we are able to initialize generation of images from other classes, allowing better exploration. On Imagenet32x32 we find that our inception and FID scores are significantly higher than the best likelihood model (PixelIQN). To show the ability to scale up to larger images, we further evaluate on ImageNet128x128, where we trained a small network (smaller than the ImageNet model in SNGAN) and find that the resultant model outperforms ACGAN but is worse than SNGAN. Our result shows that EBMs can be scaled to larger datasets, and we believe generation can be significantly improved by more parameters and increased training time.

4.2 Mode Evaluation

Next we show that EBMs exhibit modes of probability on all train and test images and do not significantly overfit to the training dataset. First, to test over-fitting, we plotted

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 5: Table of Inception and FID scores for ImageNet32x32 and CIFAR10. Quantitative numbers for ImageNet32x32 from (Ostrovski et al., 2018). (*) conditional EBM models for 128x128 are smaller than those in SNGAN.

histogram of energies for CIFAR-10 train and test dataset in Figure 10 and note almost identical curves for train and test datasets. In the Appendix, we further show that the nearest neighbor of generated images are not identical to images in the training dataset.

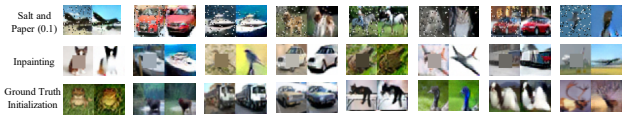


Figure 6: EBM image restoration on images in the **test** set via MCMC. The right column shows failure (approx. 10% objects change with ground truth initialization and 30% of objects change in salt/pepper corruption or in-painting. Right column shows worst case of change.)

To further test of mode coverage, we also evaluated model ability to undergo image decorruption on 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 the respective test images. In Figure 6, we find that if we initialize sampling with images from the test set, images do not move significantly, indicating modes of probability at all test images. But if the

Figure 7: 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 8: Illustration of image completions on conditional ImageNet model. Our models exhibit diversity in inpainting.

image are outside the data manifold (corrupted with noise or masked), in a large majority of cases, we are able to reliably decorrupt images, indicating relatively little mode collapse. In comparison, GANs have been shown to miss many modes of data and cannot reliably reconstruct many different test images (Yeh et al.). An advantage of implicit generation is that it allows us to directly do tasks such as decorruption without explicit knowledge of which pixels are corrupted.

As another test for mode coverage and overfitting, in Figure 8, we mask out the bottom half of ImageNet images and test the ability to regenerate the masked pixels, while clamping the value of unmasked pixels. Running Langevin dynamics on the images, we generate a diverse set of different images for both masked out train and test images, indicating relatively low overfitting on training set and diversity of modes characterized by likelihood models.

By initializing conditional models with images from images from another class, we are further able to probability modes at images far away from the those seen in training. We find in Figure 7 that surprisingly energy models are still able to reliably convert these images to images of the target class, indicating semantically meaningful modes of data even far away from training and good generalization.

4.3 Out-of-Distribution Generalization

As a test for excess modes and generalization, we evaluate the ability to detect out-of-distribution (OOD) images based off likelihood. Such a task requires both presence of high likelihood on the data manifold and low likelihood at all other locations and thus can also be seen as a proxy

(a) Illustration of images from each of the out of distribution dataset.

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

(b) AUROC scores of out of distribution classification on different datasets, only our model gets better than chance classification.

of log likelihood. Curiously, however, as found in (Nalisnick et al., 2019), it appears current likelihood models, such VAE, PixelCNN, and Glow models, are unable to distinguish data from disjoint distribution, and actually assign higher likelihood to many OOD images. We found that EBMs performed better on OOD images, and found that our proposed OOD metric correlated well with training progress – both early training and overfitting led to low OOD scores.

Following (Hendrycks and Gimpel, 2016), we propose a OOD metric using Area Under the ROC Curve (AUROC) scores computed based on classifying test dataset images from OOD images by comparing relative log likelihoods. We take unconditional generative models trained on CIFAR-10 and evaluate on CIFAR-10 test images and OOD images. We choose to evaluate on SVHN, Textures (Cimpoi et al., 2014), constant random color, uniform noise and interpolations of separate CIFAR-10 images as our of OOD distributions. We choose the SVHN dataset for comparison to previous works, Textures to test for memorization of textures, constant flat image to test for memorization of smoothness, uniform noise as a sanity test for likelihood modeling, and CIFAR-10 interpolation (where we mix two different CIFAR-10 images) to generate out of distribution images in the approximately the same domain and statistics. We provide examples of OOD images in Figure 9a.

As seen in Table 9b, unconditional EBMs perform better out-of-distribution than other unconditional models. We provide histograms of relative likelihoods for SVHN in Figure 10 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 and loss in EBMs helps eliminate spurious minima. Second, we believe EBMs have a flexible structure allowing global context when estimating probability without imposing constraints on latent variable structure. In contrast, auto-regressive models model likelihood sequentially, wehich 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, consequently making it likely that large amounts of

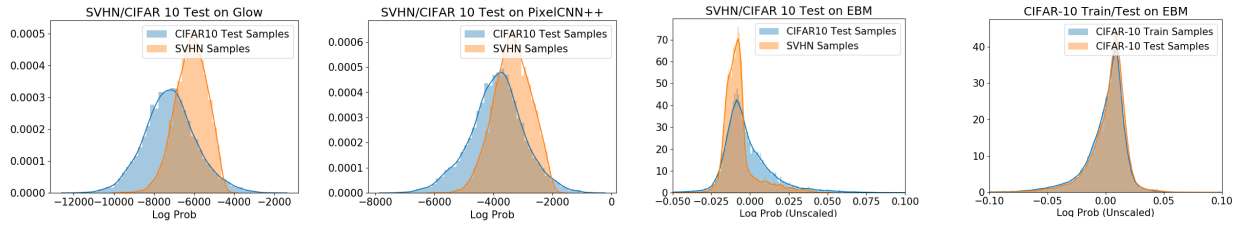
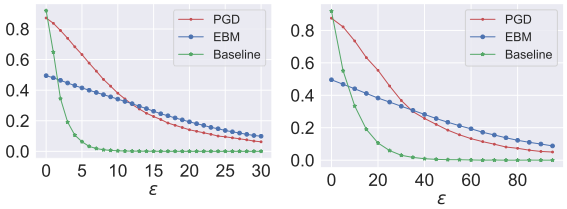


Figure 10: Histogram of relative likelihoods for various datasets for Glow, PixelCNN++ and EBM models

probability are wasted at connections between modes.

4.4 Adversarial Robustness



(a) L_∞ robustness (b) L_2 Robustness

Figure 11: 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.

To additionally test generalization, we evaluate adversarial robustness of conditional EBMs training on CIFAR-10. To compute logits for classification, we fed a given image into each class conditional model and computing the lowest energy class, without fine-tuning for classification acheiving an overall accuracy of 49.6%.

We found that classification exhibited adversarial robustness, despite a lack of adversarial training in Figure 11. 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 these 10 steps of sampling was less successful than adversarial examples on logits without sampling. Overall we find in Figure 11 that EBMs are very robust to adversarial perturbations and outperforms the SOTA L_∞ model in (Madry et al., 2017) on L_∞ attacks with $\epsilon > 13$.

5 Robotic Hand Prediction

We further show that EBMs generate and generalize well in the different domain of trajectory modeling. We train energy functions 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

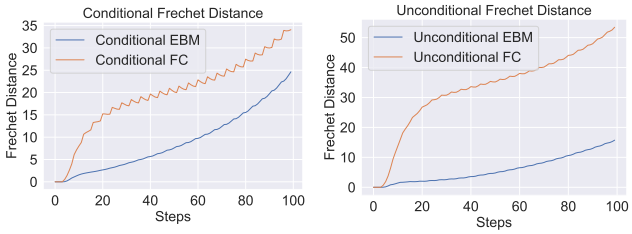


Figure 12: Conditional and Unconditional Modeling of Hand Manipulation through Frechet Distance

for diversity), with 90-10 train-test split and randomized dynamics. Models are trained to predict positions of all joints in the hand and orientation and position of the cube one step in the future. To test generalization, we evaluate the feasibility of many step roll-outs of self-predicted trajectories, since even a couple steps will enter states not seen during training.

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 feedforward models to prevent multi-step explosion. We found MSE insufficient to evaluate feasibility of long term rollouts. Even small accumulation of error quickly blow up and even after several timesteps, multistep MSE is unable to distinguish between models that simply predict the previous state compared to a model able to represent realistic dynamics.

As a result, we measure feasibility of trajectories by measuring the distance between the ground truth distribution of all states in all trajectories at timestep t as compared to that of all predicted states at time-step t across trajectories. Since there are inherent structure in states models go to, models with more accurate dynamics will have smaller distances. To compute the value, we fit a multivariate Gaussian on all different states at each specific time-step t and compute Frechet Distance (Dowson and Landau, 1982) between predicted and ground distributions, similar to the FID score proposed in (Heusel et al., 2017). Empirically, we found this metric able to distinguish better between good and bad multi-step rollouts as seen in Figure 13, where both FC and

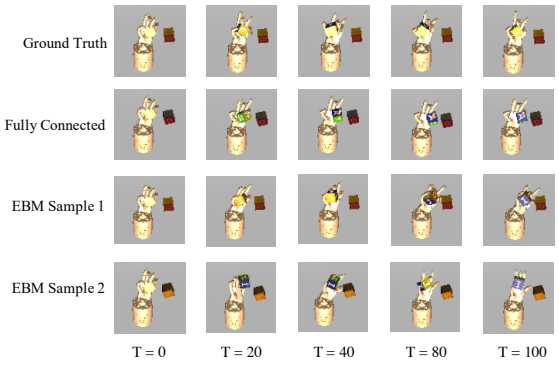


Figure 13: Top down views of robot hand manipulation trajectories generated unconditionally from the same starting state(1st frame). The FC network predicts a hand that does not move, while the EBM is able to generate distinctively different trajectories that are feasible.

EBM based models achieve similar long term MSEs despite a complete failure of the FC to model future hand dynamics, which is differentiated by our metric.

5.2 Multi-Step Trajectory Generation

As a test for generalization, 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. Overall, we find that EBM based modeling has significantly lower Frechet distance. We provide plots of Frechet distance over time in Figure 12.

In Figure 12, 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 stops predicting hand movement after several several steps as demonstrated in Figure 13. In contrast, Frechet distance increases slowly for unconditional EBMs, and we find that unconditional EBMs maintain realistic dynamics, and even models cube rotation. Furthermore, in Figure 13 we also find that unconditional EBMs are further able to generate diverse different trajectories. For conditional modeling, qualitatively we found that both conditional EBMs and FCs were able to accurately model hand movement, with the FC model diverging slightly earlier, in line with values found in Frechet distance.

6 Compositional Generalization

As a further test of generalization of EBMs, we test combining energy functions with other independent energy functions to jointly generate new samples. We construct each energy function to represent conditioning on a separate latent. Assuming independence between each conditional distribution, generation through joint conditioning on all latents is represented by generation through an energy function represented as the sum of each latent conditional energy function (Hinton, 1999) and corresponds to a product of experts model. As seen in Figure 14, summation naturally allows composition of energy functions. Furthermore, we find that this is a good test of generalization, as summation dramatically alters the sampling landscape. We sample from joint conditional distribution by undergoing a Langevin dynamic step 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 the case for (Higgins et al., 2017).

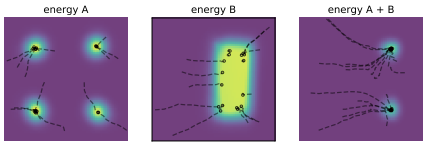


Figure 14: A 2D example of combining energy functions through their summation and the resulting sampling trajectories.

Joint Conditioning In Figure 15, we provide generated images from joint conditional sampling. We find that under joint conditional sampling we are able to generate images very close to ground truth for all classes with exception of shape. An advantage of implicit generation with energy functions is the ability for sampling from N conditional distributions in $O(N)$ time as opposed to exponentially in the case of rejection sampling. This result further shows that our models exhibit modes of likelihood on all data.

Zero-Shot Cross Product Generalization We evaluate the ability of EBMs to generalize to combinations of latents never seen before in training. We generate three datasets, D1, which consists of different size squares at a central position, D2, which consists of smallest size squares at each location, and D3, which consists of different at 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

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

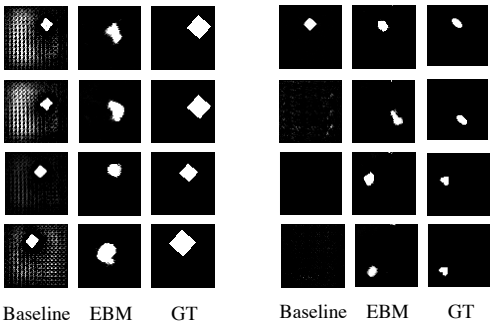


Figure 16: GT = Ground Truth. Images of cross product generalization of size-position (left panel) and shape-position (right panel).

finetune the summation energy to generate the training data using a KL term defined in the appendix). We compare against a baseline where we train a conditional model jointly conditioned on both latents.

We present results of generalization in Figure 16. In the left panel of Figure 16, we find the energy functions are able to generalize to different sizes at different position (albeit with loss in sample quality) while a conditional model ignores the size latent, generates only images seen in the training data. In the right panel of Figure 16, we found that energy functions 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 for each shape latent doesn't match the precise shape of the original shape latent), while a baseline is unable to generate samples. We believe the compositional nature of energy functions is crucial to generalize under such scenarios.

7 Conclusion

We have presented a series of techniques to scale up energy-based model training to complex high-dimensional datasets and showed that energy based models provide a number benefits, such as much sharper generation than other likelihood models or image and robot trajectory domains. Implicit generation procedures combined with energy-based models allow for compositionality and flexible denoising and inpainting. The combination also exhibits good generalization, from out-of-distribution classification to adversarial robustness, and zero-shot compositional generation. With this work, we hope to motivate more adoption of energy based models in the future.

8 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

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.

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: Nonlinear 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.

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.

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.

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.

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.

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

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.

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.

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.

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

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.

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.

A Appendix

A.1 Additional Qualitative Evaluation

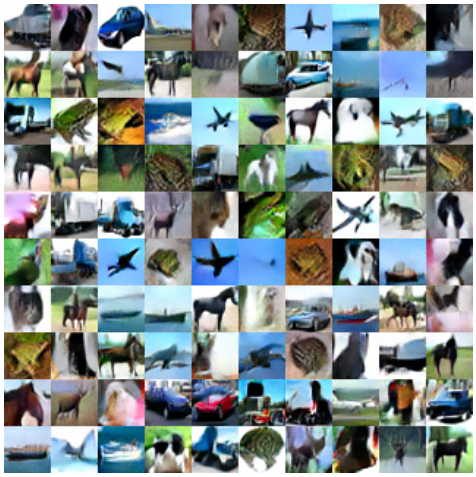


Figure 17: MCMC samples from conditional CIFAR-10 energy function

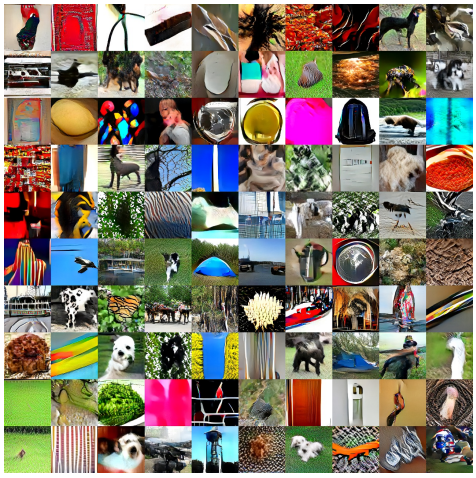


Figure 18: MCMC samples from conditional ImageNet128x128 models

We present qualitative images from a conditional generation on CIFAR10 in Figure 17 and from conditional generation of ImageNet128x128 in Figure 18, which we generate using the last 10 model snapshots of energy models. We find the presence of objects and scenes in some of the generated image with occasional hybrids (such as a presence of a toaster cat in middle bottom row).

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.

Finally, we analyze nearest neighbors of images we generate in Figure 20.

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.2 Test Time Sampling Process

We provide illustration of image generation from conditional and unconditional EBM models starting from random noise in Figure 21 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.

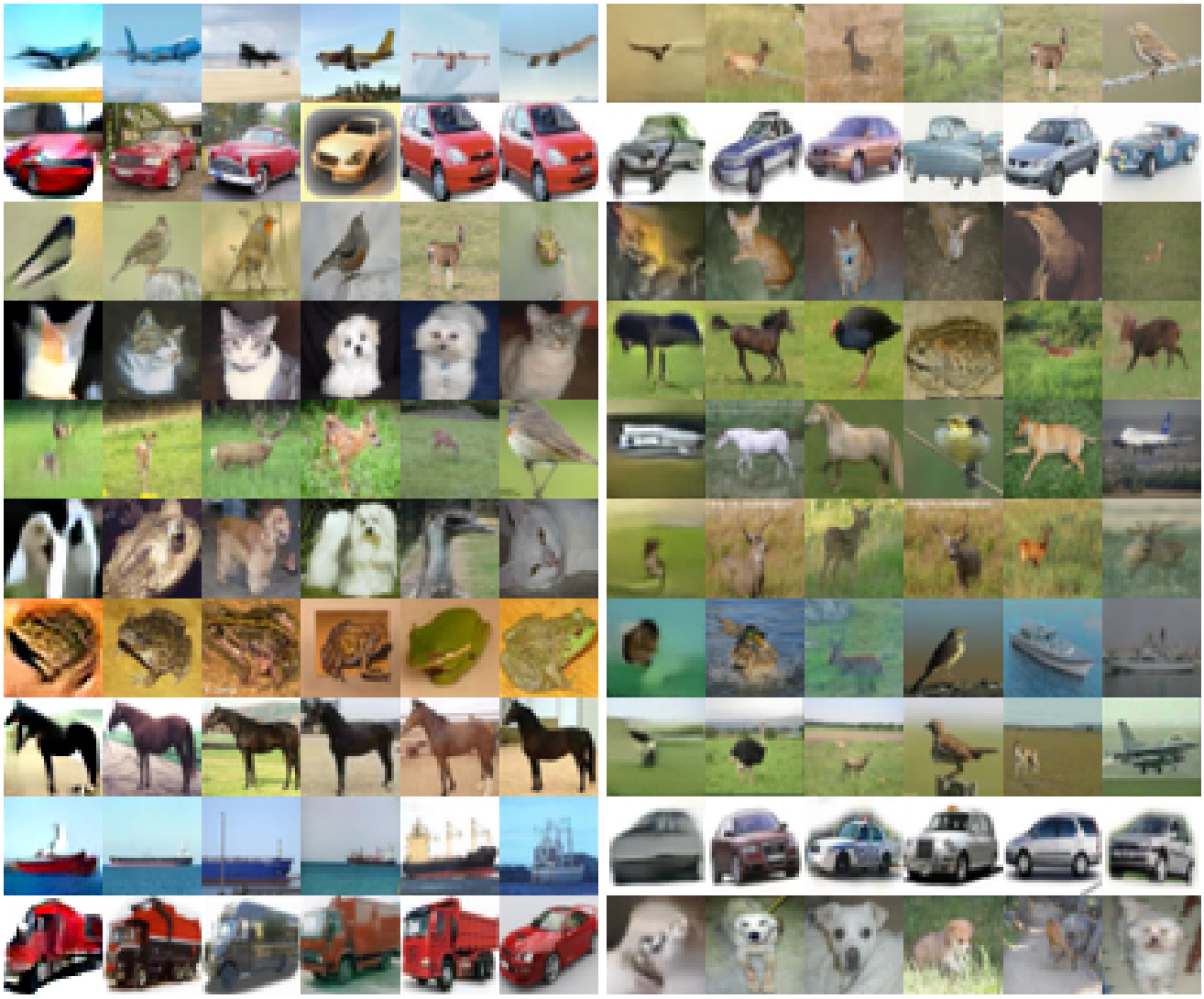
We find that if small amounts of random noise are added, all sampling procedures generate a large initial set of diverse, reduced sample quality images before converging into a small set of high probability/quality image modes that are modes of images in CIFAR10. However, we find that if sufficient noise 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.

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 0 and 1 and added 1/256 random



(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 20: Nearest neighbor images (L_2 distance) for images generated from implicit sampling.

noise following (Uria et al., 2013). Table 22 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

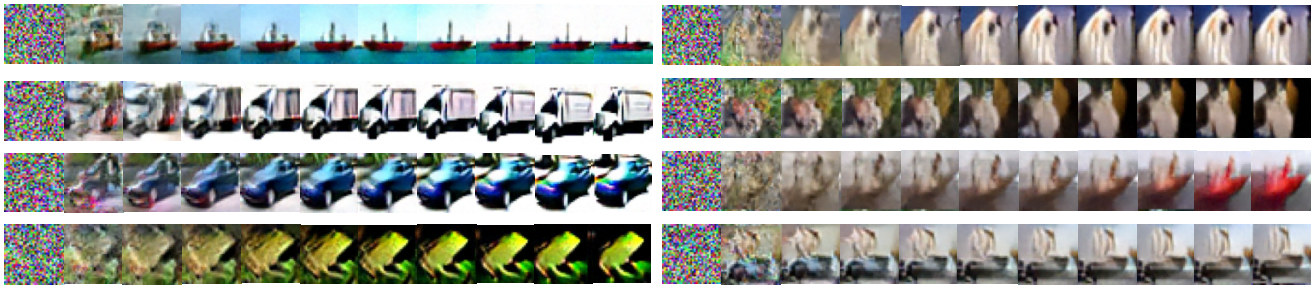
Empirically, we found that EBM training under our technique was relatively insensitive to the hyper-parameters. For example, Table 23 shows log likelihoods on continuous MNIST across several different order of magnitudes of L2 regularization and step size magnitude. We find consistent likelihood and 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).

A.5 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:

$$\mathcal{L}_{\text{KL}}(\theta) = \text{KL}(q_\theta \mid\mid p) = \mathbb{E}_{\tilde{\mathbf{x}} \sim q_\theta}\left[\bar{E}(\tilde{\mathbf{x}})\right] + \mathcal{H}\left[q_\theta\right] \quad (9)$$

Where $\bar{E}$ is treated as a constant target function that does not depend on θ. Optimizing the above loss requires differenti-



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

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

Figure 21: Generation of images from random noise.

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 22: 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 23: 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 across several different order of magnitude of variation

ating through the Langevin dynamics sampling procedure of (3), 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 $\mathcal{H}[q_\theta]$ 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.

A.6 Model

We use the residual model in Figure 24a for conditional CIFAR-10 images generation and the residual model in Figure 24b 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 24d for generation of conditional ImageNet32x32 images.

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.7 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 without backpropagating through the sampling procedure. 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.8 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.