Maximum Entropy Generators for Energy-Based Models

Rithesh Kumar[1]* **Sherjil Ozair**[1] **Anirudh Goyal**[1]
Aaron Courville[1,2] **Yoshua Bengio**[1,2,3]

[1]Mila, Université de Montréal
[2]Canadian Institute for Advanced Research (CIFAR)
[3] Institute for Data Valorization (IVADO)

Abstract

Maximum likelihood estimation of energy-based models is a challenging problem due to the intractability of the log-likelihood gradient. In this work, we propose learning both the energy function and an amortized approximate sampling mechanism using a neural generator network, which provides an efficient approximation of the log-likelihood gradient. The resulting objective requires maximizing entropy of the generated samples, which we perform using recently proposed nonparametric mutual information estimators. Finally, to stabilize the resulting adversarial game, we use a zero-centered gradient penalty derived as a necessary condition from the score matching literature. The proposed technique can generate sharp images with Inception and FID scores competitive with recent GAN techniques, does not suffer from mode collapse, and is competitive with state-of-the-art anomaly detection techniques.

1 Introduction

Unsupervised learning promises to take advantage from unlabelled data, and is regarded as crucial for artificial intelligence (Lake et al., 2017). Energy-based modeling (EBMs, LeCun et al. (2006)) is a family of unsupervised learning methods focused on learning an energy function, i.e., an unnormalized log density of the data. This removes the need to make parametric assumptions about the data distribution to make the normalizing constant (Z) tractable. However, in practice, due to the very same lack of restrictions, learning high-quality energy-based models is fraught with challenges. To avoid explicitly computing Z or its gradient, Contrastive Divergence (Hinton, 2000) and Stochastic Maximum Likelihood (Younes, 1998; Tieleman, 2008a) rely on Markov Chain Monte Carlo (MCMC) to approximately sample from the energy-based model. However, MCMC-based sampling approaches frequently suffer from long mixing times for high-dimensional data. Thus, training of energy-based models has not remained competitive with other unsupervised learning techniques such as variational auto-encoders (Kingma & Welling, 2014) and generative adversarial networks or GANs (Goodfellow et al., 2014).

In this work, we propose Maximum Entropy Generators (MEG), a framework in which we train both an energy function and an approximate sampler, which can either be fast (using a generator network G) or uses G to initialize a Markov chain in the latent space of the generator. Training such a generator properly requires entropy maximization of the generator's output distribution, for which we take advantage of recent advances in nonparametric mutual information maximization (Belghazi et al., 2018; Hjelm et al., 2018; Oord et al., 2018; Poole et al., 2018).

[*]Correspondence to rithesh.kumar@umontreal.ca

Preprint. Under review.

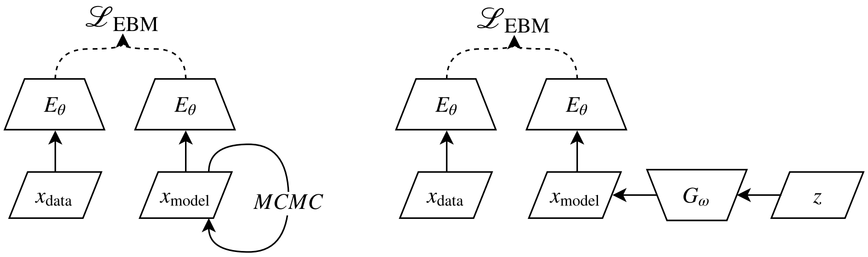


Figure 1 – **Left**: Traditional maximum likelihood training of energy-based models. **Right:** Training of maximum entropy generators for energy-based models

To evaluate the efficacy of the proposed technique, we compare against other state-of-the-art techniques on image generation, accurate mode representation, and anomaly detection. We demonstrate that the proposed technique is able to generate CIFAR-10 samples which are competitive with WGAN-GP (Gulrajani et al., 2017) according to the Fréchet Inception Distance (Heusel et al., 2017) and Inception Score (Salimans et al., 2016), and is able to generate samples of all the 10^4 modes of 4-StackedMNIST at the correct data frequencies.

We demonstrate that our technique trains energy functions useful for anomaly detection on the KDD99 dataset, and that it performs as well as state-of-the-art anomaly detection techniques which were specially designed for the task, and vastly outperform other energy-based and generative models for anomaly detection.

To summarize our contributions, we propose maximum entropy generators (MEG), a novel framework for training energy-based models using amortized neural generators and mutual information maximization. We show that the resulting energy function can be successfully used for anomaly detection, and outperforms recently published results with energy-based models. We show that MEG generates sharp images – with competitive Inception and FID scores – and accurately captures more modes than standard GANs, while not suffering from the common mode-mixing issue of many maximum likelihood generative models which results in blurry samples.

2 Background

Let $\boldsymbol{x}$ denote a sample in the data space $\mathcal{X}$ and $E_\theta : \mathcal{X} \to \mathbb{R}$ an energy function corresponding to the negative logarithm of an unnormalized estimated density density function

$$p_\theta(\boldsymbol{x}) = \frac{e^{-E_\theta(\boldsymbol{x})}}{Z_\theta} \propto e^{-E_\theta(\boldsymbol{x})} \tag{1}$$

where $Z_\theta := \int e^{-E_\theta(\boldsymbol{x})} d\boldsymbol{x}$ is the normalizing constant or partition function. Let p_D be the training distribution, from which the training set is drawn. Towards optimizing the parameters θ of the energy function, the maximum likelihood parameter gradient is

$$\frac{\partial \mathbb{E}_{\boldsymbol{x}\sim p_D}[-\log p_\theta(\boldsymbol{x})]}{\partial \theta} = \mathbb{E}_{\boldsymbol{x}\sim p_D}\left[\frac{\partial E_\theta(\boldsymbol{x})}{\partial \theta}\right] - \mathbb{E}_{\boldsymbol{x}\sim p_\theta(\boldsymbol{x})}\left[\frac{\partial E_\theta(\boldsymbol{x})}{\partial \theta}\right] \tag{2}$$

where the second term is the gradient of $\log Z_\theta$, and the sum of the two expectations is zero when training has converged, with expected energy gradients in the positive phase (under the data p_D) matching those under the negative phase (under $p_\theta(\boldsymbol{x})$). Training thus consists in trying to separate two distributions: the positive phase distribution (associated with the data) and the negative phase distribution (where the model is free-running and generating configurations by itself). This observation has motivated the pre-GAN idea presented by Bengio (2009) that "model samples are negative examples" and a classifier could be used to learn an energy function if it separated the data distribution from the model's own samples. Shortly after introducing GANs, Goodfellow (2014) also made a similar connection, related to noise-contrastive estimation (Gutmann & Hyvarinen, 2010). One should also recognize the similarity between Eq. 2 and the objective function for Wasserstein GANs or WGAN (Arjovsky et al., 2017).

The main challenge in Eq. 2 is to obtain samples from the distribution p_θ associated with the energy function E_θ. Although having an energy function is convenient to obtain a score allowing comparison of the relative probability for different $\boldsymbol{x}$'s, it is difficult to convert an energy function into a generative process. The commonly studied approaches for this are based on Markov Chain Monte Carlo, in which one iteratively updates a candidate configuration, until these configurations converge in distribution to the desired distribution p_θ. For the RBM, the most commonly used algorithms have been Contrastive Divergence (Hinton, 2000) and Stochastic Maximum Likelihood (Younes, 1998; Tieleman, 2008a), relying on the particular structure of the RBM to perform Gibbs sampling. Although these MCMC-based methods are appealing, RBMs (and their deeper form, the deep Boltzmann machine) have not been competitive in recent years compared to autoregressive models (van den Oord et al., 2016), variational auto-encoders (Kingma & Welling, 2014) and generative adversarial networks or GANs (Goodfellow et al., 2014).

What has been hypothesized as a reason for poorer results obtained with energy-based models trained with an MCMC estimator for the negative phase gradient is that running a Markov chain in data space is fundamentally difficult when the distribution is concentrated (e.g, near manifolds) and has many modes separated by vast areas of low probability. This mixing challenge is discussed by Bengio et al. (2013) who argue that a Markov chain is very likely to produce only sequences of highly probable configurations: if two modes are far from each other and only local moves are possible (which is typically the case when performing MCMC), it becomes exponentially unlikely to traverse the "desert" of low probability that can separate two modes. This makes mixing between modes difficult in high-dimensional spaces with strong concentration of probability mass in some regions (e.g. corresponding to different categories) and very low probability elsewhere.

3 Maximum Entropy Generators for Energy-Based Models

We thus propose using an amortized neural sampler to perform fast approximate sampling to train the energy model. We begin by replacing the model distribution p_θ in in Eq. 2 by a neural generator G parametrized by $\boldsymbol{w}$. We define P_G as the distribution of the outputs $G(z)$ for $\boldsymbol{z} \sim p_z$ where p_z is a simple prior distribution such as a standard Normal distribution.

$$\frac{\partial \mathcal{L}_E}{\partial \theta} = \mathbb{E}_{\boldsymbol{x}\sim p_D}\left[\frac{\partial E_\theta(\boldsymbol{x})}{\partial \theta}\right] - \mathbb{E}_{\boldsymbol{x}\sim p_G(\boldsymbol{x})}\left[\frac{\partial E_\theta(\boldsymbol{x})}{\partial \theta}\right] \tag{3}$$

To minimize the approximation error, p_G must be close to p_θ. To do so, we tune G to minimize the KL divergence $KL(p_G||p_\theta)$, which can be rewritten in terms of minimizing the energy of the samples from the generator while maximizing the entropy at the output of the generator:

$$\begin{aligned} KL(p_G||p_\theta) &= -H[p_G] - E_{p_G}[\log p_\theta(\boldsymbol{x})] & (4)\\ &= -H[p_G] + E_{p_G}[E_\theta(\boldsymbol{x})] + \log Z_\theta & (5)\end{aligned}$$

When taking the gradient of $KL(p_G||p_\theta)$ with respect to the parameters $\boldsymbol{w}$ of the generator, the log-partition function $\log Z_\theta$ disappears and we can optimize $\boldsymbol{w}$ by minimizing

$$\mathcal{L}_G = -H[p_G] + \mathbb{E}_{\boldsymbol{z}\sim p_z} E_\theta(G(\boldsymbol{z})) \tag{6}$$

where p_z is the prior distribution of the latent variable of the generator.

In order to approximately maximize the entropy $H[p_G]$ at the output of the generator, we use one recently proposed nonparametric mutual information maximization techniques (Belghazi et al., 2018; Oord et al., 2018; Hjelm et al., 2018). Poole et al. (2018) show that these techniques can be unified into a single framework derived from the variational bound of Barber & Agakov (2003). Since the generator is deterministic, mutual information between inputs and outputs reduces to simply entropy of the outputs, since the conditional entropy of a deterministic function is zero:

$$I(X, Z) = H(X) - H(X|Z) = H(G(Z)) - \cancelto{0}{H(G(Z)|Z)}$$

In particular, we use the estimator from Hjelm et al. (2018), which estimates the Jensen-Shannon divergence between the joint distribution ($p(\boldsymbol{x}, \boldsymbol{z})$) and the product of marginals ($p(\boldsymbol{x})p(\boldsymbol{z})$). We refer to this information measure as $I_{JSD}(X, Z)$. We found that the JSD-based estimator works better in practice than the KL-based estimator (which corresponds to the mutual information).

The estimator of Hjelm et al. (2018) is given by

$$\mathcal{I}_{JSD}(X, Z) = \sup_{T\in\mathcal{T}} \mathbb{E}_{p(X,Z)}[-\mathrm{sp}(-T(X, Z))] - \mathbb{E}_{p(X)p(Z)}[\mathrm{sp}(T(X, Z))] \tag{7}$$

where $\mathrm{sp}(a) = \log(1 + e^a)$ is the softplus function. The supremum is approximated using gradient descent on the parameters of the discriminator T.

With $X = G(Z)$ the output of the generator, $\mathcal{I}_{JSD}(G(Z), Z)$ is one of the terms to be maximized in the objective function for training G, which would maximize the generator's output entropy $H(G(Z))$.

Thus the final training objective to be minimized for the generator G and the energy function E is

$$\mathcal{L}_G = -\mathcal{I}_{JSD}(G(Z), Z) + \mathbb{E}_{\boldsymbol{z} \sim p_z} E_\theta(G(\boldsymbol{z})) \tag{8}$$

$$\mathcal{L}_E = \mathbb{E}_{\boldsymbol{x} \sim p_D} E_\theta(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{z} \sim p_z} E_\theta(G(\boldsymbol{z})) \tag{9}$$

where $Z \sim p_z$, the latent prior (typically a $N(0, I)$ Gaussian).

3.1 Improving training stability

As can be seen from the above equations, the generator and the energy function are in an adversarial game, similar to generative adversarial networks (Goodfellow et al., 2014). This makes optimization via simultaneous gradient descent challenging since the gradient vector field of such an optimization problem is non-conservative as noted by Mescheder et al. (2017). This is particularly accentuated by the use of deep neural networks for the generator and the energy function. In particular, we noticed that during training the magnitude of the energy function values would diverge.

To help alleviate this issue we look towards another technique for learning energy-based models called *score matching* proposed by Hyvärinen (2005). Score matching estimates the energy function by matching the score functions of the data density and the model density, where the score function ψ is the gradient of the log density with respect to the sample $\psi(\boldsymbol{x}) = \frac{\partial \log p(\boldsymbol{x})}{\partial \boldsymbol{x}}$. If $\psi_D(\boldsymbol{x})$ and $\psi_E(\boldsymbol{x})$ are the score functions under the data distribution and model distribution respectively, the score matching objective is given by

$$\mathcal{J}_{SM} = \mathbb{E}_{\boldsymbol{x} \sim P_{\mathrm{D}}} \left[\|\psi_D(\boldsymbol{x}) - \psi_E(\boldsymbol{x})\|_2^2 \right].$$

While the score function for the data distribution is typically unknown and would require estimation, Theorem 1 in Hyvärinen (2005) shows that with partial integrations, the score matching objective can be reduced to the following objective which does not depend on the score function under the data distribution:

$$\begin{aligned}
\mathcal{J}_{SM} &= \mathbb{E}_{\boldsymbol{x} \sim P_{\mathrm{D}}} \left[\sum_i \partial_i \psi_i(\boldsymbol{x}) + \frac{1}{2} \psi_i(\boldsymbol{x})^2 \right] \\
&= \mathbb{E}_{\boldsymbol{x} \sim P_{\mathrm{D}}} \left[\sum_i -\frac{\partial^2 E(\boldsymbol{x})}{\partial^2 \boldsymbol{x}_i} + \frac{1}{2} \left(\frac{-\partial E(\boldsymbol{x})}{\partial \boldsymbol{x}_i} \right)^2 \right] \\
&= \mathbb{E}_{\boldsymbol{x} \sim P_{\mathrm{D}}} \left[\frac{1}{2} \left\| \frac{\partial E(\boldsymbol{x})}{\partial \boldsymbol{x}} \right\|_2^2 + \sum_i -\frac{\partial^2 E(\boldsymbol{x})}{\partial^2 \boldsymbol{x}_i} \right] \qquad (10)
\end{aligned}$$

The above objective is hard to optimize when using deep neural networks because of the difficulty in estimating the gradient of the Hessian diagonal, so we use the first term in our objective, i.e. the zero-centered gradient penalty, pushing the data points to sit near critical points (generally a local minimum) of the energy function.

This term is also similar to the gradient penalty regularization proposed by Gulrajani et al. (2017) which however is one-centered and applied on interpolations of the data and model samples, and is derived from the Lipschitz continuity requirements of Wasserstein GANs (Arjovsky et al., 2017).

3.2 Improving sample quality via latent space MCMC

Since MEG simultaneously trains a generator and a valid energy function, we can improve the quality of samples by biasing sampling towards high density regions. Furthermore, doing the MCMC walk in the latent space should be easier than in data space because the transformed data manifold (in latent space) is flatter than in the original observed data space, as initially discussed by Bengio et al.

(2013). The motivation is also similar to that of the "truncation trick" used successfully by Brock et al. (2018). However, we use an MCMC-based approach for this which is applicable to arbitrary latent distributions.

We use the Metropolis-adjusted Langevin algorithm (MALA, Girolami & Calderhead (2011)), with Langevin dynamics producing a proposal distribution in the latent space as follows:

$$\tilde{\mathbf{z}}_{t+1} = \boldsymbol{z}_t - \alpha \frac{\partial E_\theta(G_\omega(\boldsymbol{z}_t))}{\partial \boldsymbol{z}_t} + \epsilon\sqrt{2 * \alpha}, \text{ where } \epsilon \sim \mathcal{N}(0, I_d)$$

Next, the proposed $\tilde{z}_{t+1}$ is accepted or rejected using the Metropolis Hastings algorithm, by computing the acceptance ratio:

$$r = \frac{p(\tilde{\mathbf{z}}_{t+1})q(\mathbf{z}_t|\tilde{\mathbf{z}}_{t+1})}{p(\mathbf{z}_t)q(\tilde{\mathbf{z}}_{t+1}|\mathbf{z}_t)} \tag{11}$$

$$\frac{p(\tilde{\mathbf{z}}_{t+1})}{p(\mathbf{z}_t)} = \exp\left\{ -E_\theta(G_\omega(\tilde{\mathbf{z}}_{t+1})) + E_\theta(G_\omega(\mathbf{z}_t))\right\} \tag{12}$$

$$q(\tilde{\mathbf{z}}_{t+1}|\mathbf{z}_t) \propto \exp\left(\frac{-1}{4\alpha}||\tilde{\mathbf{z}}_{t+1} - \mathbf{z}_t + \alpha \frac{\partial E_\theta(G_\omega(\mathbf{z}_t))}{\partial \mathbf{z}_t}||_2^2\right) \tag{13}$$

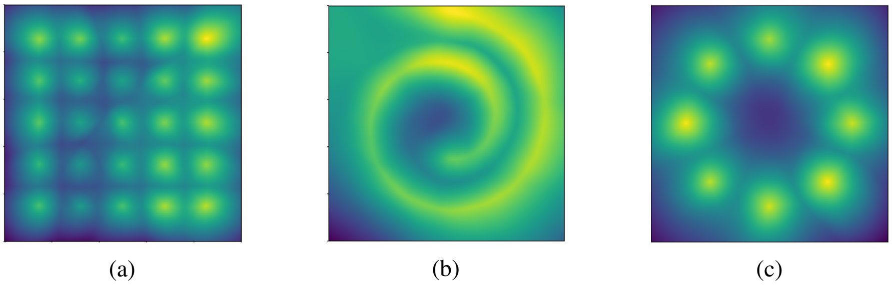
and accepting (setting $\boldsymbol{z}_{t+1} = \tilde{\boldsymbol{z}}_{t+1}$) with probability r.

4 Experiments

To understand the benefits of MEG, we first visualize the energy densities learnt by our generative model on toy data. Next, we evaluate the efficacy of our entropy maximizer by running discrete mode collapse experiments to verify that we learn all modes and the corresponding mode count (frequency) distribution. Furthermore, we evaluate the performance of MEG on sharp image generation, since this is a common failure mode of models trained with maximum likelihood which tend to generate blurry samples (Theis et al., 2015). We also compare MCMC samples in visible space and our proposed sampling from the latent space of the composed energy function. Finally, we run anomaly detection experiments to test the application of the learnt energy function.

We've released open-source code[2] for all the experiments.

4.1 Visualizing the learned energy function


(a)

(b)

(c)

Figure 2 – Probability density visualizations for three popular toy dataset (a) 25-gaussians, (b) swiss roll, and (c) 8-gaussians. Density was estimated using a sample based approximation of the partition function.

Generative models trained with maximum likelihood often suffer from the problem of spurious modes and excessive entropy of the trained distribution, where the model incorrectly assigns high probability mass to regions not present in the data manifold. Typical energy-based models such as RBMs suffer from this problem partly because of the poor approximation of the negative phase gradient, as discussed above, and the large price paid in terms of log-likelihood for not putting enough probability mass near data points (i.e. for missing modes).

[2] https://github.com/ritheshkumar95/energy_based_generative_models

To check if MEG suffers from spurious modes, we train the energy-based model on synthetic 2D datasets (swissroll, 25gaussians and 8gaussians) and visualize the energy function. From the probability density plots on Figure 2, we can see that the energy model doesn't suffer from spurious modes and learns a sharp distribution.

4.2 Investigating Mode Collapse

GANs are notorious for having mode collapse issues wherein certain modes of the data distribution are not represented by the generated distribution. Since the generator is trained to minimize its KL divergence with the energy model distribution (which is trained via maximum likelihood), we expect the generator to faithfully capture all the modes of the data distribution. Our theory requires we maximize entropy of the generated distribution, which we believe is instrumental in ensuring full mode capture.

To empirically verify MEG captures all the modes of the data distribution, we follow the same experimental setup as (Metz et al., 2016) and (Srivastava et al., 2017). We train our generative model on the StackedMNIST dataset, which is a synthetic dataset created by stacking MNIST on different channels. The number of modes are counted using a pretrained MNIST classifier, and the KL divergence is calculated empirically between the generated mode distribution and the data distribution.

Table 1 – Number of captured modes and Kullback-Leibler divergence between the training and samples distributions for ALI (Dumoulin et al., 2016), Unrolled GAN (Metz et al., 2016), VeeGAN (Srivastava et al., 2017), WGAN-GP (Gulrajani et al., 2017). Numbers except MEG and WGAN-GP are borrowed from Belghazi et al. (2018)

(Max 10^3)	Modes	KL
Unrolled GAN	48.7	4.32
VEEGAN	150.0	2.95
WGAN-GP	959.0	0.7276
MEG (ours)	1000.0	0.0313

(Max 10^4)	Modes	KL
WGAN-GP	9538.0	0.9144
MEG (ours)	10000.0	0.0480

From Table 1, we can see that MEG naturally covers all the modes in that data, without dropping a single mode. Apart from just representing all the modes of the data distribution, MEG also better matches the data distribution as evidenced by the significantly smaller KL divergence score compared to the baseline WGAN-GP.

Apart from the standard 3-StackMNIST, we also evaluate MEG on a new dataset with 10^4 modes (4 stacks)[3] which is evidence that MEG does not suffer from mode collapse issues unlike state-of-the-art GANs like WGAN-GP.

4.3 Modeling Natural Images

While the energy landscapes in Figure 2 provide evidence that MEG trains energy models with sharp distributions, we next investigate if this also holds when learning a distribution over high-dimensional natural images. Energy-based models trained with existing techniques produce blurry samples due to the energy function not learning a sharp distribution.

We train MEG on the standard benchmark 32x32 CIFAR10 dataset for image modeling. We additionally train MEG on the 64x64 cropped CelebA - celebrity faces dataset to report qualitative samples from MEG. Similar to recent GAN works (Miyato et al., 2018), we report both Inception Score (IS) and Fréchet Inception Distance (FID) scores on the CIFAR10 dataset and compare it with a competitive WGAN-GP baseline.

[3] The 4-StackedMNIST was created in a way analogous to the original 3-StackedMNIST dataset. We randomly sample and fix 128×10^4 images to train the generative model and take 26×10^4 samples for evaluations.

Table 2 – Inception scores and FIDs with unsupervised image generation on CIFAR-10. We used 50000 sample estimates to compute Inception Score and FID.

Method	**Inception score**	**FID**
Real data	11.24±.12	7.8
WGAN-GP	6.81 ± .08	30.95
MEG (Generator)	6.49 ± .05	35.02
MEG (MCMC)	**7.31 ± .06**	33.18

From Table 2, we can see that in addition to learning an energy function, MEG-trained generative model produces samples comparable to recent GAN methods such as WGAN-GP (Gulrajani et al., 2017). Note that the perceptual quality of the samples improves by using the proposed MCMC sampler in the latent space. See also Figure 3 in Appendix B for an ablation study which shows that MCMC on the visible space does not perform as well as MCMC on the latent space.

4.4 Anomaly Detection

Apart from the usefulness of energy estimates for relative density estimation (up to the normalization constant), energy functions can also be useful to perform unsupervised anomaly detection. Unsupervised anomaly detection is a fundamental problem in machine learning, with critical applications in many areas, such as cyber-security, complex system management, medical care, etc. Density estimation is at the core of anomaly detection since anomalies are data points residing in low probability density areas. We test the efficacy of our energy-based density model for anomaly detection using two popular benchmark datasets: KDDCUP and MNIST.

KDDCUP We first test our generative model on the KDDCUP99 10 percent dataset from the UCI repository (Lichman et al., 2013). Our baseline for this task is Deep Structured Energy-based Model for Anomaly Detection (DSEBM) (Zhai et al., 2016), which trains deep energy models such as Convolutional and Recurrent EBMs using denoising score matching (Vincent, 2011) instead of maximum likelihood, for performing anomaly detection. We also report scores on the state of the art DAGMM (Zong et al., 2018), which learns a Gaussian Mixture density model (GMM) over a low dimensional latent space produced by a deep autoencoder. We train MEG on the KDD99 data and use the score norm $||\nabla_x E_\theta(x)||_2^2$ as the decision function, similar to Zhai et al. (2016).

Table 3 – Performance on the KDD99 dataset. Values for OC-SVM, DSEBM values were obtained from Zong et al. (2018). Values for MEG are derived from 5 runs. For each individual run, the metrics are averaged over the last 10 epochs.

Model	Precision	Recall	F1
Kernel PCA	0.8627	0.6319	0.7352
OC-SVM	0.7457	0.8523	0.7954
DSEBM-e	0.8619	0.6446	0.7399
DAGMM	0.9297	0.9442	0.9369
MEG (ours)	**0.9354 ± 0.016**	**0.9521 ± 0.014**	**0.9441 ± 0.015**

From Table 3, we can see that the MEG energy function outperforms the previous SOTA energy-based model (DSEBM) by a large margin (+0.1990 F1 score) and is comparable to the current SOTA model (DAGMM) specifically designed for anomaly detection. Note that DAGMM is specially designed for anomaly detection, while MEG is a general-purpose energy-based model.

MNIST Next we evaluate our generative model on anomaly detection of high dimensional image data. We follow the same experiment setup as (Zenati et al., 2018) and make each digit class an anomaly and treat the remaining 9 digits as normal examples. We also use the area under the precision-recall curve (AUPRC) as the metric to compare models. From Table 4, it can be seen that our energy model outperforms VAEs for outlier detection and is comparable to the SOTA BiGAN-based anomaly detection methods for this dataset (Zenati et al., 2018) which train bidirectional GANs to learn both

Table 4 – Performance on the unsupervised anomaly detection task on MNIST measured by area under precision recall curve. Numbers except ours are obtained from (Zenati et al., 2018). Results for MEG are averaged over the last 10 epochs to account for the variance in scores.

Heldout Digit	VAE	MEG	BiGAN-σ
1	0.063	0.281 ± 0.035	0.287 ± 0.023
4	0.337	0.401 ± 0.061	0.443 ± 0.029
5	0.325	0.402 ± 0.062	0.514 ± 0.029
7	0.148	0.29 ± 0.040	0.347 ± 0.017
9	0.104	0.342 ± 0.034	0.307 ± 0.028

an encoder and decoder (generator) simultaneously and use a combination of the reconstruction error in output space as well as the discriminator's cross entropy loss as the decision function.

Our aim here is not to claim state-of-the-art on the task of anomaly detection but to demonstrate the quality of the energy functions learned by our technique, as judged by its competitive performance on anomaly detection.

5 Related Work

Early work on deep learning relied on unsupervised learning (Hinton et al., 2006; Bengio et al., 2007; Larochelle et al., 2009) to train energy-based models (LeCun et al., 2006), in particular Restricted Boltzmann Machines, or RBMs. Hinton (2000) proposed k-step Contrastive Divergence (CD-k), to efficiently approximate the negative phase log-likelihood gradient. Subsequent work have improved on CD-k such as Persistent CD (Salakhutdinov & Hinton, 2009; Tieleman, 2008b). Hyvärinen (2005) proposed an alternative method to train non-normalized graphical models using Score Matching, which does not require computation of the partition function.

Kim & Bengio (2016) and Dai et al. (2017) also learn a generator that approximates samples from an energy-based model. However, their approach for entropy maximization is different from our own. Kim & Bengio (2016) argue that batch normalization (Ioffe & Szegedy, 2015) makes the hidden activations of the generator network approximately Gaussian distributed and thus maximize the log-variance for each hidden activation of the network. Dai et al. (2017) propose two approaches to entropy maximization. One which minimizes entropy of the inverse model ($p_{gen}(z|x)$) which is approximated using an amortized inverse model similar to ALI (Dumoulin et al., 2016), and another which makes isotropic Gaussian assumptions for the data. In our work, we perform entropy maximization using a tight mutual information estimator which does not make any assumptions about the data distribution.

Zhao et al. (2016) use an autoencoder as the discriminator and use the reconstruction loss as a signal to classify between real and fake samples. The autoencoder is highly regularized to allow its interpretation as an energy function. However Dai et al. (2017) prove that the EBGAN objective does not guarantee the discriminator to recover the true energy function. The generator diverges from the true data distribution after matching it, since it would continue to receive training signal from the discriminator. The discriminator signal does not vanish even at optimality (when $P_G = P_D$) if it retains density information, since some samples would be considered "more real" than others.

6 Conclusion

We proposed MEG, an energy-based generative model that produces energy estimates using an energy model and a generator that produces fast approximate samples. This takes advantage of novel methods to maximize the entropy at the output of the generator using a nonparametric mutual information lower bound estimator. We have shown that our energy model learns good energy estimates using visualizations in toy 2D datasets and through performance in unsupervised anomaly detection. We have also shown that our generator produces samples of high perceptual quality by measuring Inception Scores and Fréchet Inception Distance and shown that MEG is robust to the respective weaknesses of GAN models (mode dropping) and maximum-likelihood energy-based models (spurious modes).

7 Acknowledgements

The authors acknowledge the funding provided by NSERC, Canada AI Chairs, Samsung, Microsoft, Google and Facebook. We also thank NVIDIA for donating a DGX-1 computer used for certain experiments in this work.

References

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

Barber, D. and Agakov, F. The im algorithm: A variational approach to information maximization. In *Proceedings of the 16th International Conference on Neural Information Processing Systems*, NIPS'03, pp. 201–208, Cambridge, MA, USA, 2003. MIT Press. URL `http://dl.acm.org/citation.cfm?id=2981345.2981371`.

Belghazi, I., Baratin, A., Rajeswar, S., Ozair, S., Bengio, Y., Courville, A., and Hjelm, R. D. Mine: mutual information neural estimation. *arXiv preprint arXiv:1801.04062, ICML'2018*, 2018.

Bengio, Y. *Learning deep architectures for AI*. Now Publishers, 2009.

Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. Greedy layer-wise training of deep networks. In *NIPS'2006*, 2007.

Bengio, Y., Mesnil, G., Dauphin, Y., and Rifai, S. Better mixing via deep representations. In *ICML'2013*, 2013.

Brock, A., Donahue, J., and Simonyan, K. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Dai, Z., Almahairi, A., Bachman, P., Hovy, E., and Courville, A. Calibrating energy-based generative adversarial networks. *arXiv preprint arXiv:1702.01691*, 2017.

Dumoulin, V., Belghazi, I., Poole, B., Mastropietro, O., Lamb, A., Arjovsky, M., and Courville, A. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

Girolami, M. and Calderhead, B. Riemann manifold langevin and hamiltonian monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214, 2011.

Goodfellow, I. On distinguishability criteria for estimating generative models. *arXiv preprint arXiv:1412.6515*, 2014.

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

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. Improved training of Wasserstein GANs. In *NIPS'2017*, pp. 5767–5777, 2017.

Gutmann, M. and Hyvarinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS'2010*, 2010.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pp. 6626–6637, 2017.

Hinton, G. E. Training products of experts by minimizing contrastive divergence. Technical Report GCNU TR 2000-004, Gatsby Unit, University College London, 2000.

Hinton, G. E., Osindero, S., and Teh, Y. W. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–1554, 2006.

Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Trischler, A., and Bengio, Y. Learning deep representations by mutual information estimation and maximization. *arXiv:1808.06670*, 2018.

Hyvärinen, A. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(Apr):695–709, 2005.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Kim, T. and Bengio, Y. Deep directed generative models with energy-based probability estimation. In *ICLR 2016 Workshop, arXiv:1606.03439*, 2016.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017.

Larochelle, H., Bengio, Y., Louradour, J., and Lamblin, P. Exploring strategies for training deep neural networks. *J. Machine Learning Res.*, 10:1–40, 2009.

LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M.-A., and Huang, F.-J. A tutorial on energy-based learning. In Bakir, G., Hofman, T., Scholkopf, B., Smola, A., and Taskar, B. (eds.), *Predicting Structured Data*, pp. 191–246. MIT Press, 2006.

Lichman, M. et al. Uci machine learning repository, 2013.

Mescheder, L., Nowozin, S., and Geiger, A. The numerics of gans. In *Advances in Neural Information Processing Systems*, pp. 1825–1835, 2017.

Metz, L., Poole, B., Pfau, D., and Sohl-Dickstein, J. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*, 2016.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Poole, B., Ozair, S., Oord, A. v. d., Alemi, A., and Tucker, G. On variational lower bounds of mutual information. 2018.

Salakhutdinov, R. and Hinton, G. Deep boltzmann machines. In *Artificial intelligence and statistics*, pp. 448–455, 2009.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training gans. In *Advances in neural information processing systems*, pp. 2234–2242, 2016.

Srivastava, A., Valkoz, L., Russell, C., Gutmann, M. U., and Sutton, C. Veegan: Reducing mode collapse in gans using implicit variational learning. In *Advances in Neural Information Processing Systems*, pp. 3308–3318, 2017.

Theis, L., Oord, A. v. d., and Bethge, M. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2015.

Tieleman, T. Training restricted Boltzmann machines using approximations to the likelihood gradient. In *ICML'2008*, pp. 1064–1071, 2008a.

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

van den Oord, A., Kalchbrenner, N., and Kavukcuoglu, K. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.

Vincent, P. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

Younes, L. On the convergence of Markovian stochastic algorithms with rapidly decreasing ergodicity rates. In *Stochastics and Stochastics Models*, pp. 177–228, 1998.

Zenati, H., Foo, C. S., Lecouat, B., Manek, G., and Chandrasekhar, V. R. Efficient gan-based anomaly detection. *arXiv preprint arXiv:1802.06222*, 2018.

Zhai, S., Cheng, Y., Lu, W., and Zhang, Z. Deep structured energy based models for anomaly detection. *arXiv preprint arXiv:1605.07717*, 2016.

Zhao, J., Mathieu, M., and LeCun, Y. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016.

Zong, B., Song, Q., Min, M. R., Cheng, W., Lumezanu, C., Cho, D., and Chen, H. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. 2018.

Appendix A Training Algorithm

Algorithm 1 **MEG Training Procedure** Default values: Adam parameters $\alpha = 0.0001, \beta_1 = 0.5, \beta_2 = 0.9; \lambda = 0.1; n_\varphi = 5$

```
Require: Score penalty coefficient $\lambda$, # of $\theta$ updates per generator update $n_\varphi$ , # of training iterations $T$,
  Adam hyperparameters $\alpha$, $\beta_1$ and $\beta_2$.
Require: Energy function $E_\theta$ with parameters $\theta$, entropy statistics function $T_\phi$ with parameters $\phi$, generator
  function $G_\omega$ with parameters $\omega$, minibatch size $m$,
  for $t = 1, ..., T$ do
    for $1, ..., n_\varphi$ do
      Sample minibatch of real data $\{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(m)}\} \sim P_D$.
      Sample minibatch of latent $\{\mathbf{z}_0^{(1)}, ..., \mathbf{z}_0^{(m)}\} \sim P_z$.
      $\tilde{\mathbf{x}} \leftarrow G_\omega(\mathbf{z})$
      $\mathcal{L}_E \leftarrow \frac{1}{m}\Big[\sum_i^m E_\theta(\mathbf{x}^{(i)}) - \sum_i^m E_\theta(\tilde{\mathbf{x}}^{(i)}) + \lambda \sum_i^m \big|\big|\nabla_{\mathbf{x}^{(i)}} E_\theta(\mathbf{x}^{(i)})\big|\big|^2\Big]$
      $\theta \leftarrow \text{Adam}(\mathcal{L}_E, \theta, \alpha, \beta_1, \beta_2)$
    end for
    Sample minibatch of latent $\mathbf{z} = \{\mathbf{z}^{(1)}, ..., \mathbf{z}^{(m)}\} \sim P_z$.
    Per-dimension shuffle of $\mathbf{z}$, yielding $\{\tilde{\mathbf{z}}^{(1)}, ..., \tilde{\mathbf{z}}^{(m)}\}$.
    $\tilde{\mathbf{x}} \leftarrow G_\omega(\mathbf{z})$
    $\mathcal{L}_H \leftarrow \dfrac{1}{m}\sum_i^m \Big[\log \sigma(T_\phi(\tilde{\mathbf{x}}^{(i)}, \mathbf{z}^{(i)})) - \log\big(1 - \sigma(T_\phi(\tilde{\mathbf{x}}^{(i)}, \tilde{\mathbf{z}}^{(i)}))\big)\Big]$
    $\mathcal{L}_G \leftarrow \dfrac{1}{m}\Big[\sum_i^m E_\theta(\tilde{\mathbf{x}}^{(i)})\Big] + \mathcal{L}_H$
    $\omega \leftarrow \text{Adam}(\mathcal{L}_G, \omega, \alpha, \beta_1, \beta_2)$
    $\phi \leftarrow \text{Adam}(\mathcal{L}_H, \phi, \alpha, \beta_1, \beta_2)$
  end for
```

Appendix B MCMC sampling in visible vs latent space

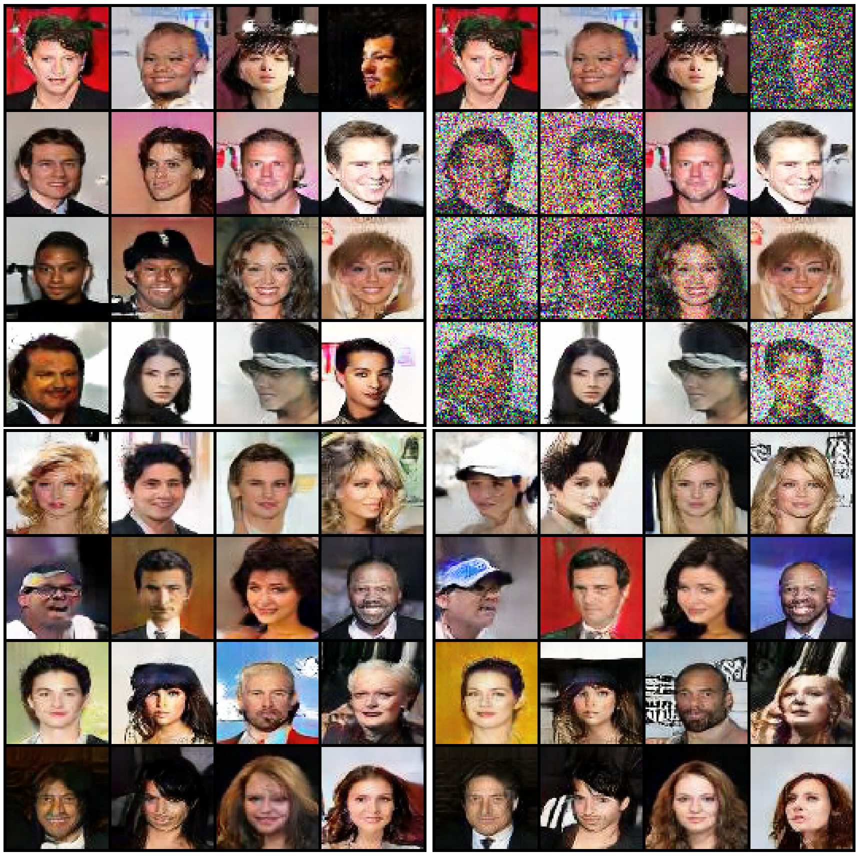


Figure 3 – Samples from the beginning and end of the MCMC in visible space (**top**) and latent space **bottom** using the MALA proposal and acceptance criteria. MCMC in visible space has poor mixing and gets attracted to spurious modes, while MCMC in latent space seems to change semantic attributes of the image, while not producing spurious modes.