Deep Generative Model

Energy Based Model

(EBM)

Ying Shi
2020.01.14

Outline:

- Discriminative model VS Generative Model
- Sampling Method in Generative Model
- Energy Based Model (EBM) : Start From RMB (Restricted Boltzmann machine, RBM)
- Implicit Generation with EBM
- Maximum Entropy Generators for EBM
- Flow & EBM

Discriminative model VS Generative Model

Discriminative model:

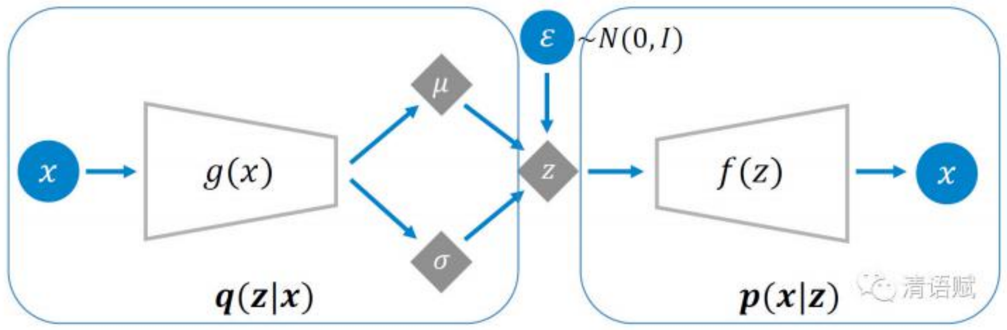
- Feature extraction
- Classifier

Generative model:

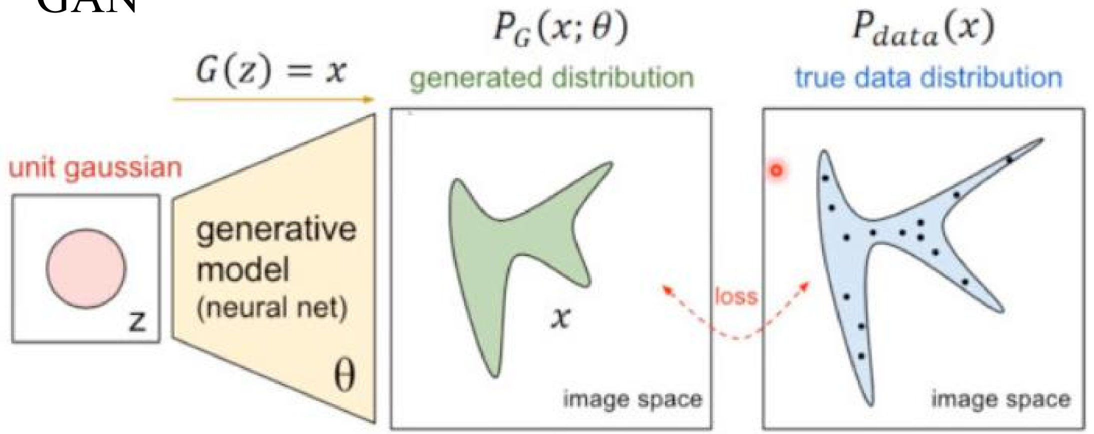
- Sampling
- Inference

Sampling Method in Generative Model

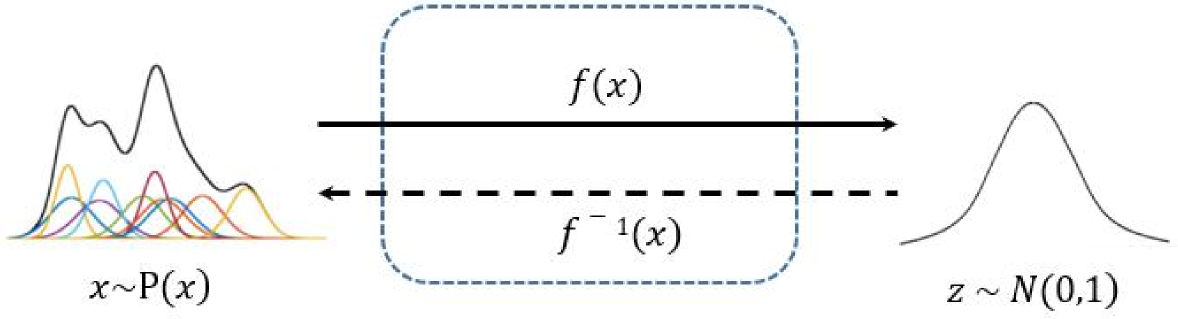
VAE



GAN



FLOW



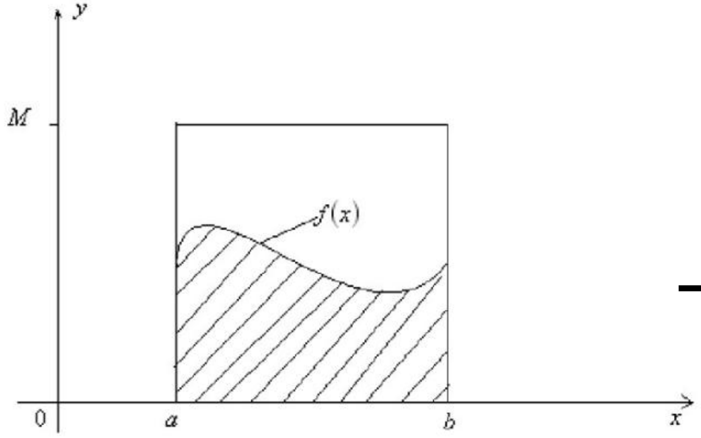
Sample from a specific distribution

Sampling Method in Generative Model

Monte Carlo Markov Chain (MCMC): MC

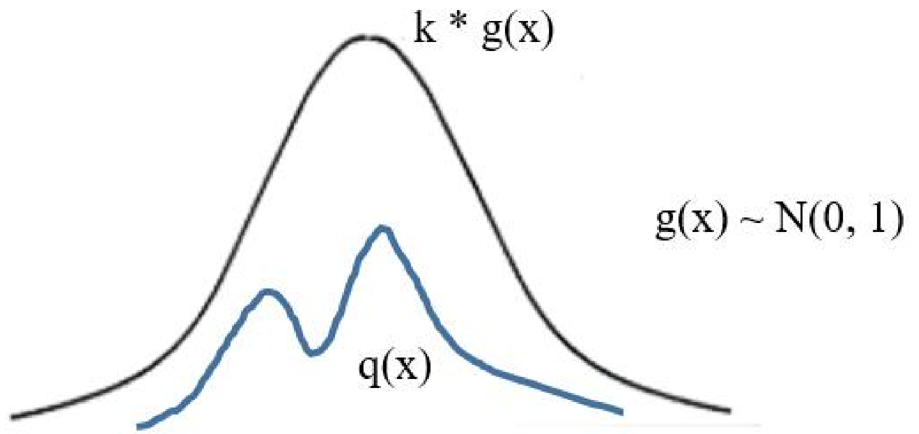
Q: We have a probability density function(PDF) which is very complex. We want some sample point from such(PDF) to analyze some statistical Characteristics. How to sample.

Monte Carlo (MC):



$$\longrightarrow \int_a^b f(x)dx = \frac{b-a}{n}\sum_{i=1}^{n} f(x_i) \longrightarrow \int_a^b \frac{f(x)}{q(x)}q(x)dx \approx \sum_{i=1}^{n}\frac{f(x)}{q(x)}$$

q(x) is hard to compute: Accept & Reject



Sampling Method in Generative Model

Monte Carlo Markov Chain (MCMC): MC

Markov Chain (MC):

$$P(X_{t+1} = s_j | X_0 = s_{i_0}, X_1 = s_{i_1}, \cdots, X_t = s_i) = P(X_{t+1} = s_j | X_t = s_i),$$ t: times; s:state

$$P_{i,j} = P(X_{t+1} = s_j | X_t = s_i).$$

$$\lim_{t \to \infty} \pi^{(0)} \mathbf{P}^t = \pi^*,$$ π^0: init state (Markov Chain should be aperiodic)

$$\pi(i)P(i,j) = \pi(j)P(j,i), \quad for\ all\ i,j$$

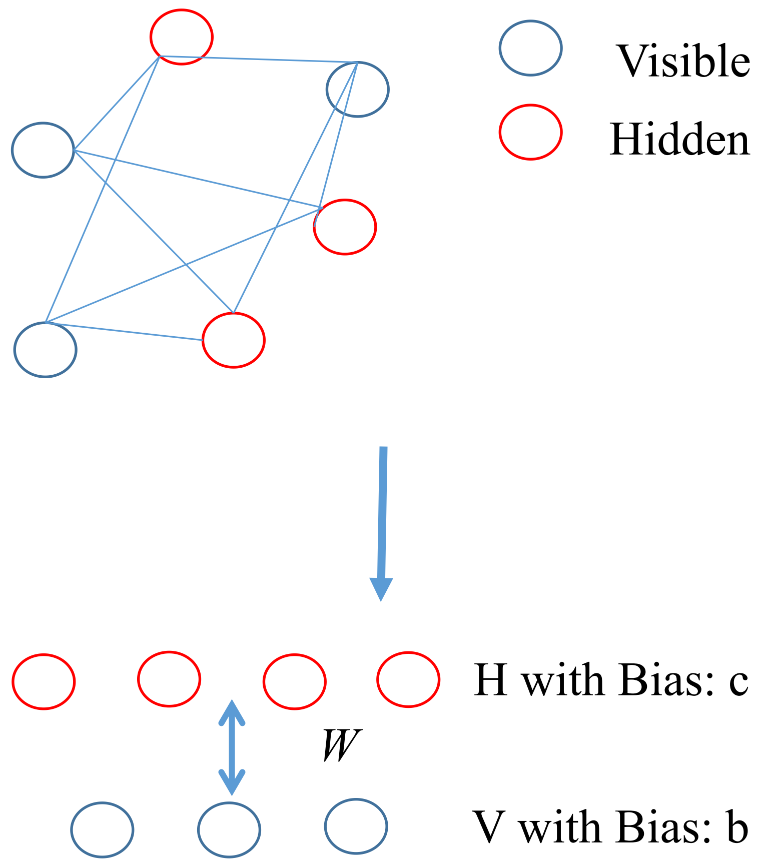
$$\pi(i)Q(i,j)\alpha(i,j) = \pi(j)Q(j,i)\alpha(j,i)$$ Inspired by Monte Carlo α(i,j) is accept & reject

if we want to do sample under a certain distribution, we just simulate the Markov process with stable distribution, after enough steps(times) of transfer, our sample distribution will be very close to the stable distribution

MCMC: M-H, Gibbs

Restricted Boltzmann machine

RBM



$$p(v,h) = \frac{1}{Z}\exp\Big(-E(v,h)\Big)$$ (A joint distribution about v & h)

$$Z = \sum_v \sum_h \exp\Big(-E(v,h)\Big) \quad E(v,h) = -b^T v - c^T h - v^T W h$$

$$\begin{aligned}
\ell(W,b,c) &= \sum_{t=1}^{n} \log P(v^{(t)}) \\
&= \sum_{t=1}^{n} \log \sum_h P(v^{(t)}, h) \\
&= \sum_{t=1}^{n} \log \sum_h \frac{1}{Z}\exp\Big(-E(v^{(t)},h)\Big) \\
&= \sum_{t=1}^{n} \log \frac{1}{Z} \sum_h \exp\Big(-E(v^{(t)},h)\Big) \\
&= \sum_{t=1}^{n} \log \sum_h \exp\Big(-E(v^{(t)},h)\Big) - \sum_{t=1}^{n} \log Z \\
&= \sum_{t=1}^{n} \log \sum_h \exp\Big(-E(v^{(t)},h)\Big) - n\log Z \\
&= \sum_{t=1}^{n} \log \sum_h \exp\Big(-E(v^{(t)},h)\Big) - n\log \sum_{v,h} \exp\Big(-E(v,h)\Big)
\end{aligned}$$

Restricted Boltzmann machine

RBM

$$\ell(W,b,c) = \sum_{t=1}^{n} \log \sum_{h} \exp\Big(-E(v^{(t)},h)\Big) - n \log \sum_{v,h} \exp\Big(-E(v,h)\Big)$$

$$\theta = \{b, c, W\} \quad \nabla_\theta \ell(W,b,c)$$

$$\nabla_\theta \ell(W,b,c) = \nabla_\theta \sum_{t=1}^{n} \log \sum_{h} \exp\Big(-E(v^{(t)},h)\Big) - n \nabla_\theta \log \sum_{v,h} \exp\Big(-E(v,h)\Big)$$

$$= \sum_{t=1}^{n} \frac{\nabla_\theta \sum_h \exp\Big(-E(v^{(t)},h)\Big)}{\sum_h \exp\Big(-E(v^{(t)},h)\Big)} - n \frac{\nabla_\theta \sum_{v,h} \exp\Big(-E(v,h)\Big)}{\sum_{v,h} \exp\Big(-E(v,h)\Big)}$$

$$= \sum_{t=1}^{n} \frac{\sum_h \exp\Big(-E(v^{(t)},h)\Big) \nabla_\theta \Big(-E(v^{(t)},h)\Big)}{\sum_h \exp\Big(-E(v^{(t)},h)\Big)} - n \frac{\sum_{v,h} \exp\Big(-E(v,h)\Big) \nabla_\theta \Big(-E(v,h)\Big)}{\sum_{v,h} \exp\Big(-E(v,h)\Big)}$$

$$P(h|v^{(t)}) = \frac{\exp\Big(-E(v^{(t)},h)\Big) / Z}{\sum_h \exp\Big(-E(v^{(t)},h)\Big) / Z}, P(h,v) = \frac{\exp\Big(-E(v,h)\Big)}{\sum_{v,h} \exp\Big(-E(v,h)\Big)},$$

$$\nabla_\theta \ell(W,b,c) = \sum_{t=1}^{n} \mathbb{E}_{P(h|v^{(t)})} \Big[\nabla_\theta \Big(-E(v^{(t)},h)\Big)\Big] - n \mathbb{E}_{P(h,v)} \Big[\nabla_\theta \Big(-E(v,h)\Big)\Big]$$

Restricted Boltzmann machine

RBM

$$\nabla_\theta \ell(W,b,c) = \sum_{t=1}^{n} \mathbb{E}_{P(h|v^{(t)})}\left[\nabla_\theta\Big(-E(v^{(t)},h)\Big)\right] - n\mathbb{E}_{P(h,v)}\left[\nabla_\theta\Big(-E(v,h)\Big)\right]$$

$$\big\downarrow$$

$$Z = \sum_v \sum_h \exp\Big(-E(v,h)\Big)$$

Def: Gibbs Steps: k, samples per batch: t

```
for(i=0; i<k+t; i++)
    $p(v^{i+1}|h^{i})$
    $p(h^{i+1}|v^{i+1})$
samples{$(v,h)^{k+1}$, $(v,h)^{k+2}$ , $(v,h)^{k+t}$}
```

$$\mathbb{E}_{P(h,v)}\left[\nabla_\theta\Big(-E(v,h)\Big)\right] \approx \nabla_\theta\Big(-E(v,h)\Big)\Big|_{v=\hat{v},h=\hat{h}}$$

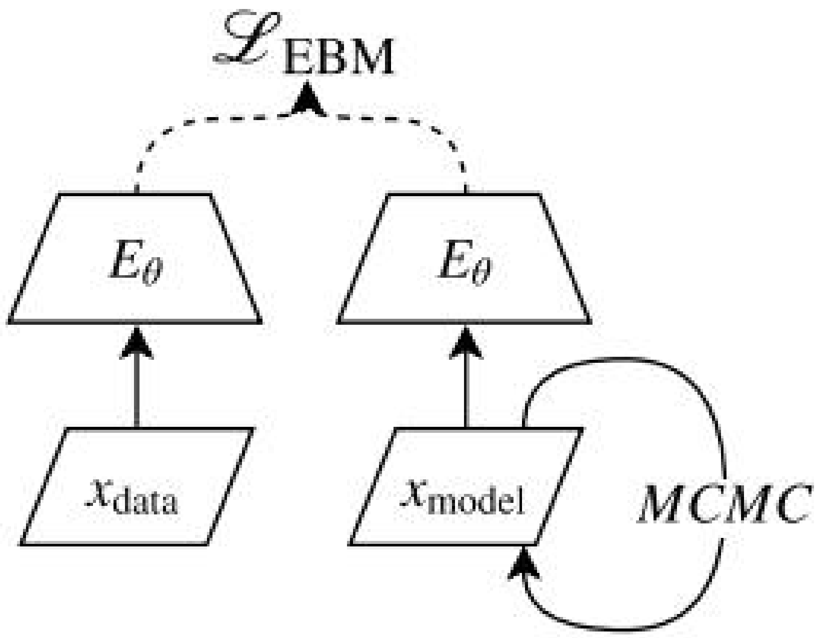
$$W = W + \alpha\Big(h(v^{(t)}){v^{(t)}}^T - h(\tilde{v})\tilde{v}^T\Big)$$

$$b = b + \alpha\Big(h(v^{(t)}) - h(\tilde{v})\Big)$$

$$c = c + \alpha\Big(v^{(t)} - \tilde{v}\Big)$$

Restricted Boltzmann machine

RBM



Implicit Generation with EBM

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

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

Sample Replay Buffer

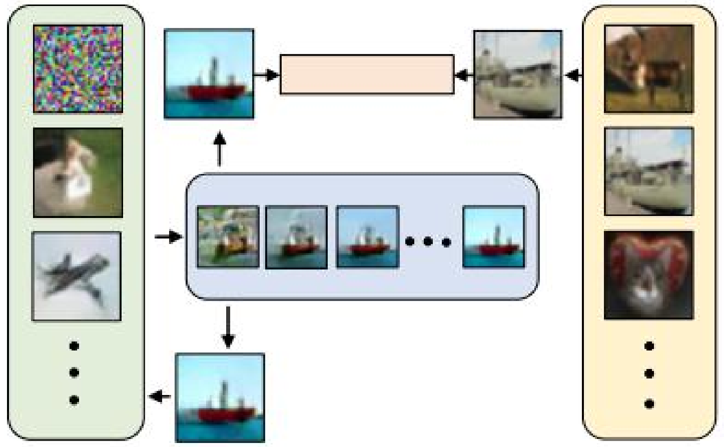


Figure 1: Overview of our method and the interrelationship of the components involved.

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. (PCD)

Du, Y., & Mordatch, I. (2019). Implicit Generation and Modeling with Energy Based Models. In *Advances in Neural Information Processing Systems* (pp. 3603-3613).

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

Implicit Generation with EBM

Algorithm 1 Energy training algorithm

```
Input: data dist. $p_D(\mathbf{x})$, step size $\lambda$, 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 $q_\theta$ via Langevin dynamics:
    for sample step $k = 1$ to $K$ do
        $\tilde{\mathbf{x}}^k \leftarrow \tilde{\mathbf{x}}^{k-1} - \nabla_{\mathbf{x}} E_\theta(\tilde{\mathbf{x}}^{k-1}) + \omega, \quad \omega \sim \mathcal{N}(0, \sigma)$
    end for
    $\mathbf{x}_i^- = \Omega(\tilde{\mathbf{x}}_i^k)$
    ▷ Optimize objective $\alpha\mathcal{L}_2 + \mathcal{L}_{ML}$ wrt $\theta$:
    $\Delta\theta \leftarrow \nabla_\theta \frac{1}{N} \sum_i \alpha(E_\theta(\mathbf{x}_i^+)^2 + E_\theta(\mathbf{x}_i^-)^2) + E_\theta(\mathbf{x}_i^+) - E_\theta(\mathbf{x}_i^-)$
    Update $\theta$ based on $\Delta\theta$ using Adam optimizer
    $\mathcal{B} \leftarrow \mathcal{B} \cup \tilde{\mathbf{x}}_i$
end while
```

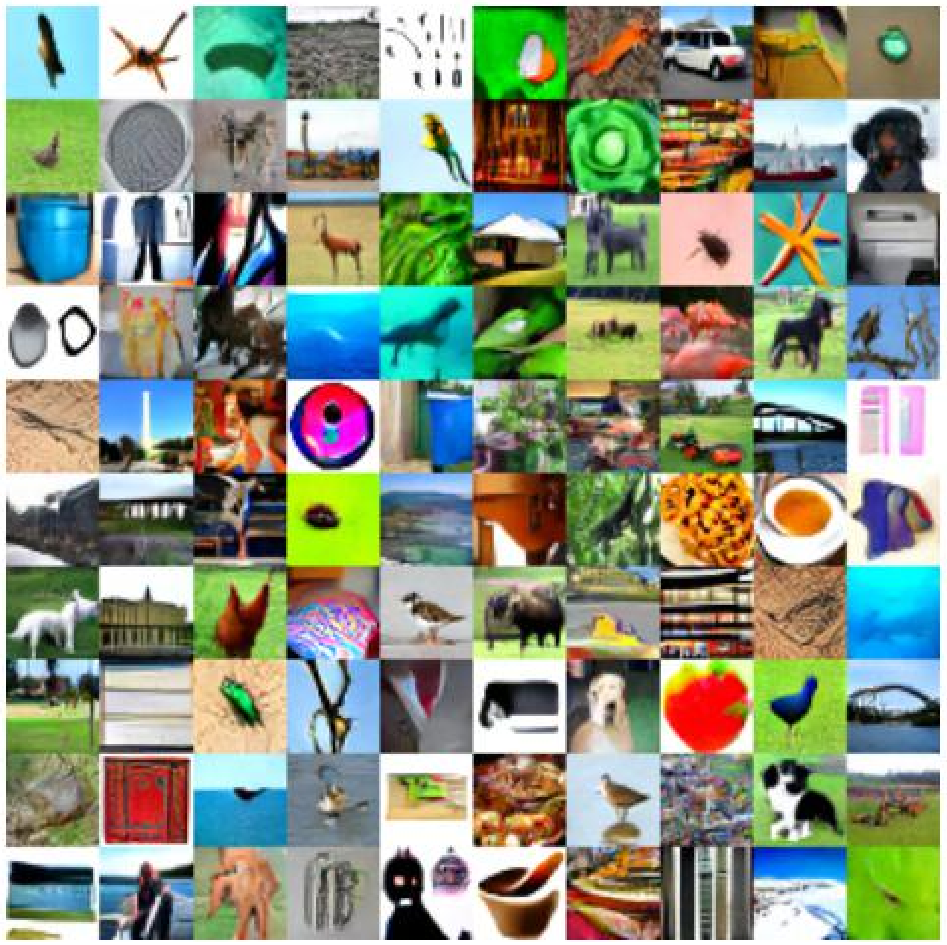


Figure 2: Conditional ImageNet32x32 EBM samples

Implicit Generation with EBM

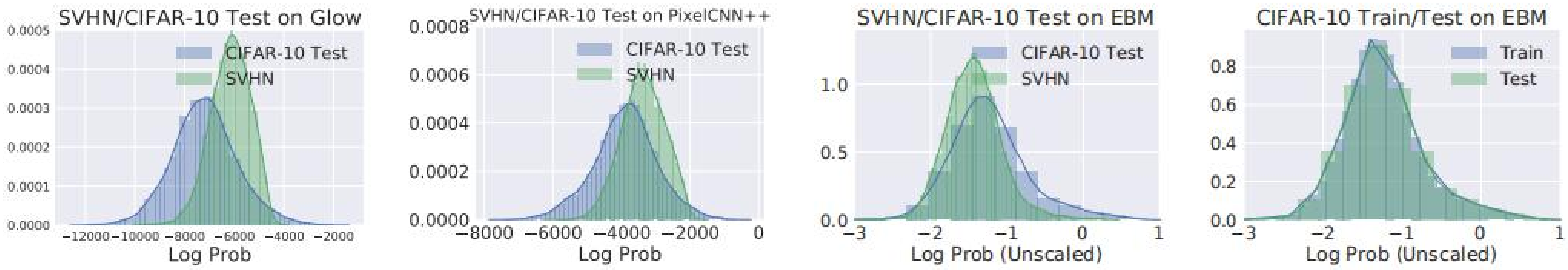


Figure 11: Histogram of relative likelihoods for various datasets for Glow, PixelCNN++ and EBM models

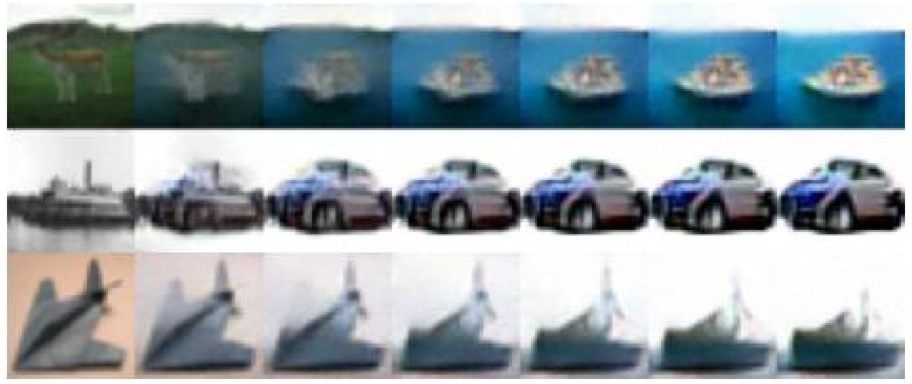


Figure 6: Illustration of cross-class implicit sampling on a conditional EBM. The EBM is conditioned on a particular class but is initialized with an image from a separate class.

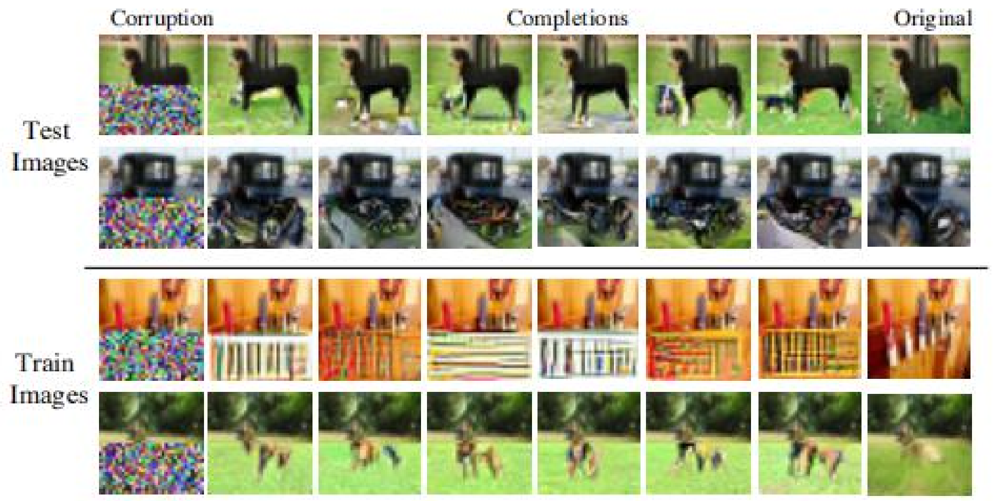


Figure 7: Illustration of image completions on conditional ImageNet model. Our models exhibit diversity in inpainting.

Implicit Generation with EBM

4.4 Out-of-Distribution Generalization

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

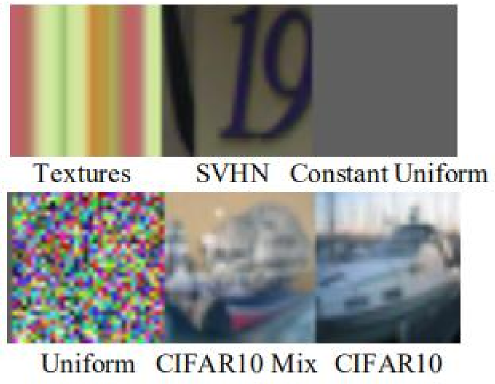
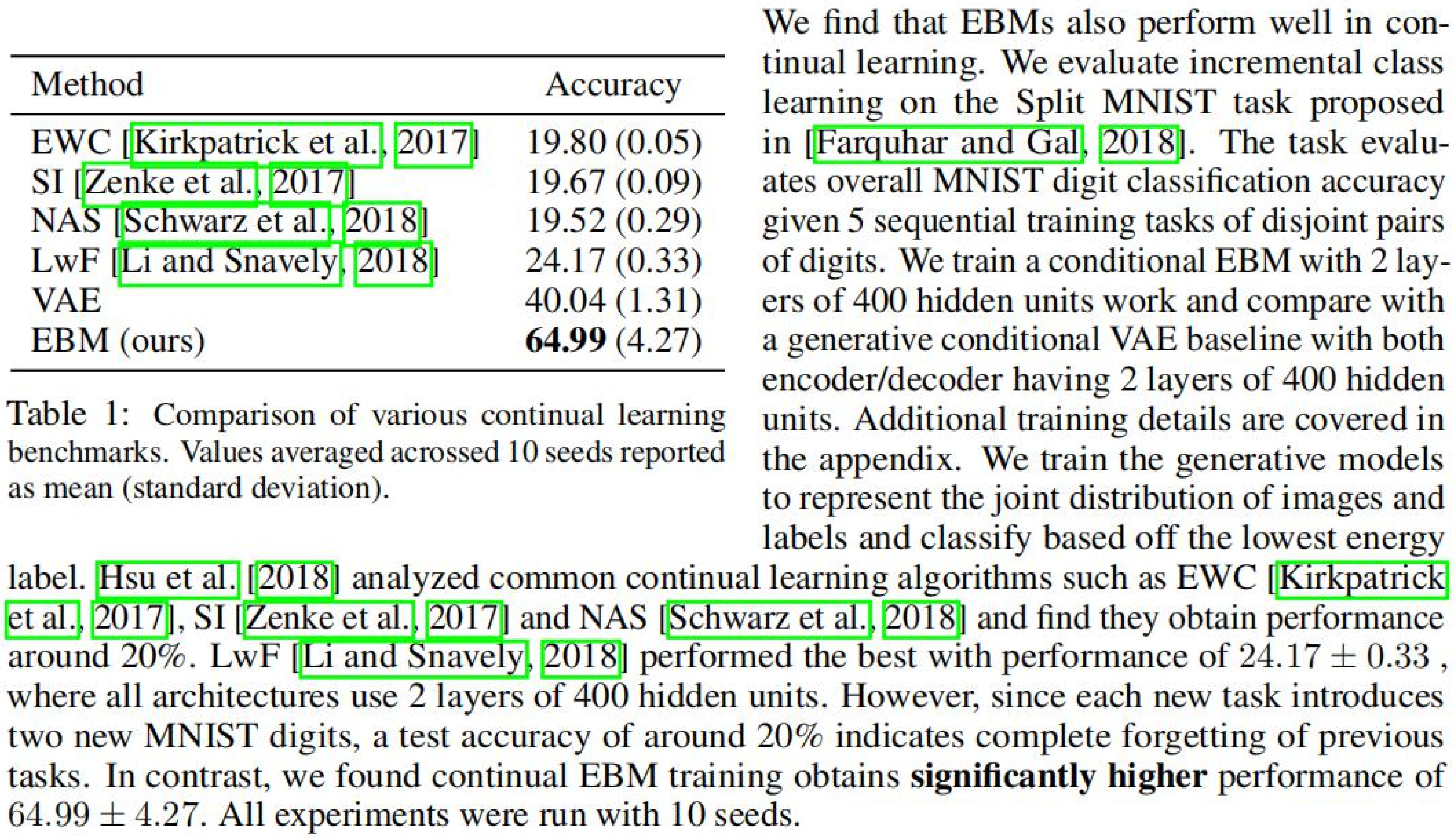


Figure 9: Illustration of images from each of the out of distribution dataset.

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

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

Implicit Generation with EBM



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

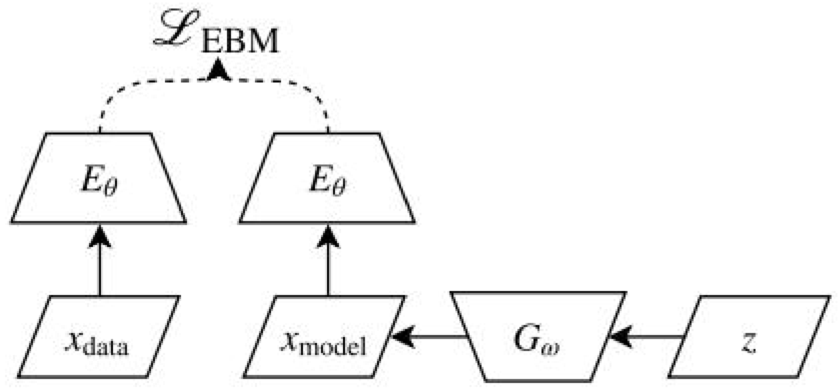
Table 1: Comparison of various continual learning benchmarks. Values averaged acrossed 10 seeds reported as mean (standard deviation).

We find that EBMs also perform well in continual learning. We evaluate incremental class learning on the Split MNIST task proposed in [Farquhar and Gal, 2018]. The task evaluates overall MNIST digit classification accuracy given 5 sequential training tasks of disjoint pairs of digits. We train a conditional EBM with 2 layers of 400 hidden units work and compare with a generative conditional VAE baseline with both encoder/decoder having 2 layers of 400 hidden units. Additional training details are covered in the appendix. We train the generative models to represent the joint distribution of images and labels and classify based off the lowest energy label. Hsu et al. [2018] analyzed common continual learning algorithms such as EWC [Kirkpatrick et al., 2017], SI [Zenke et al., 2017] and NAS [Schwarz et al., 2018] and find they obtain performance around 20%. LwF [Li and Snavely, 2018] performed the best with performance of 24.17 ± 0.33 , where all architectures use 2 layers of 400 hidden units. However, since each new task introduces two new MNIST digits, a test accuracy of around 20% indicates complete forgetting of previous tasks. In contrast, we found continual EBM training obtains **significantly higher** performance of 64.99 ± 4.27. All experiments were run with 10 seeds.

Maximum Entropy Generators for EBM

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

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



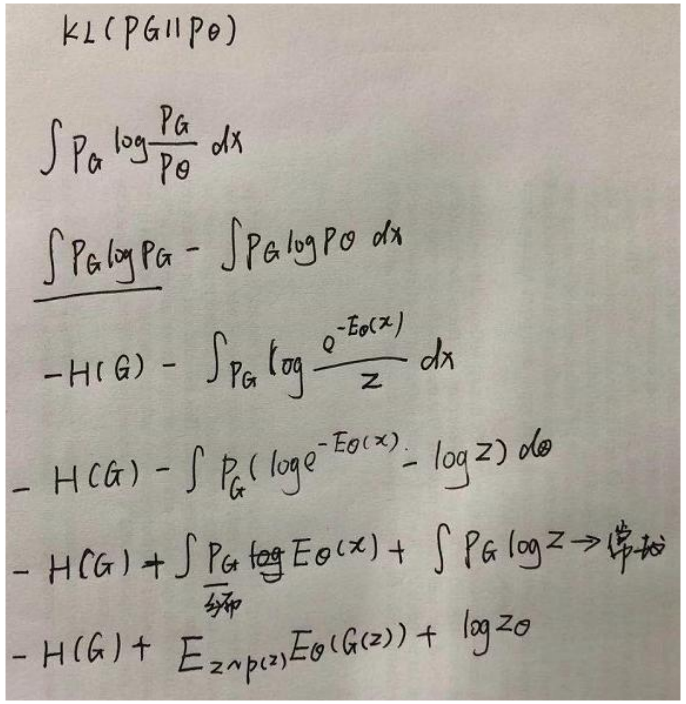
Belghazi, I., Baratin, A., Rajeswar, S., Ozair, S., Bengio, Y., Courville, A., and Hjelm, R. D. Mine: mutual information neural estimation. *arXiv preprint arXiv:1801.04062, ICML'2018*, 2018.
Girolami, M. and Calderhead, B. Riemann manifold langevin and hamiltonian monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214, 2011.
Kumar, R., Goyal, A., Courville, A., & Bengio, Y. (2019). Maximum Entropy Generators for Energy-Based Models. *arXiv preprint arXiv:1901.08508*.

Maximum Entropy Generators for EBM

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:

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

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



$KL(P_G\|P_\theta)$

$\int P_G \log \frac{P_G}{P_\theta} dx$

$\underline{\int P_G \log P_G} - \int P_G \log P_\theta \, dx$

$-H(G) - \int P_G \log \frac{e^{-E_\theta(x)}}{Z} dx$

$-H(G) - \int P_G (\log e^{-E_\theta(x)} - \log Z) \, d\theta$

$-H(G) + \int \underset{分布}{\underline{P_G}}\, \cancel{\log} E_\theta(x) + \int P_G \log Z \rightarrow$ 常数

$-H(G) + E_{z\sim p(z)} E_\theta(G(z)) + \log Z_\theta$

$$\begin{aligned}
I(X;Y) &= \int_X\int_Y P(X,Y)\log\frac{P(X,Y)}{P(X)P(Y)} \\
&= \int_X\int_Y P(X,Y)\log\frac{P(X,Y)}{P(X)} - \int_X\int_Y P(X,Y)\log P(Y) \\
&= \int_X\int_Y P(X)P(Y|X)\log P(Y|X) - \int_Y \log P(Y)\int_X P(X,Y) \\
&= \int_X P(X)\int_Y P(Y|X)\log P(Y|X) - \int_Y \log P(Y)P(Y) \\
&= -\int_X P(X)H(Y|X=x) + H(Y) \\
&= H(Y) - H(Y|X)
\end{aligned}$$

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.

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.

Maximum Entropy Generators for EBM

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

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

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

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

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

Maximum Entropy Generators for EBM

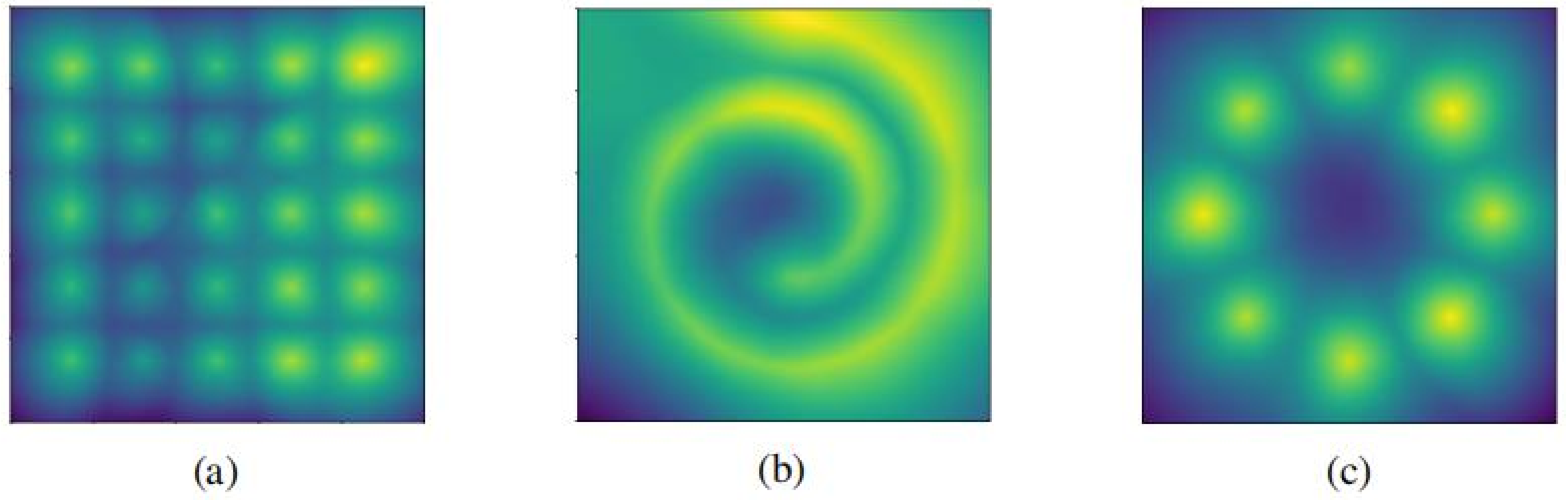
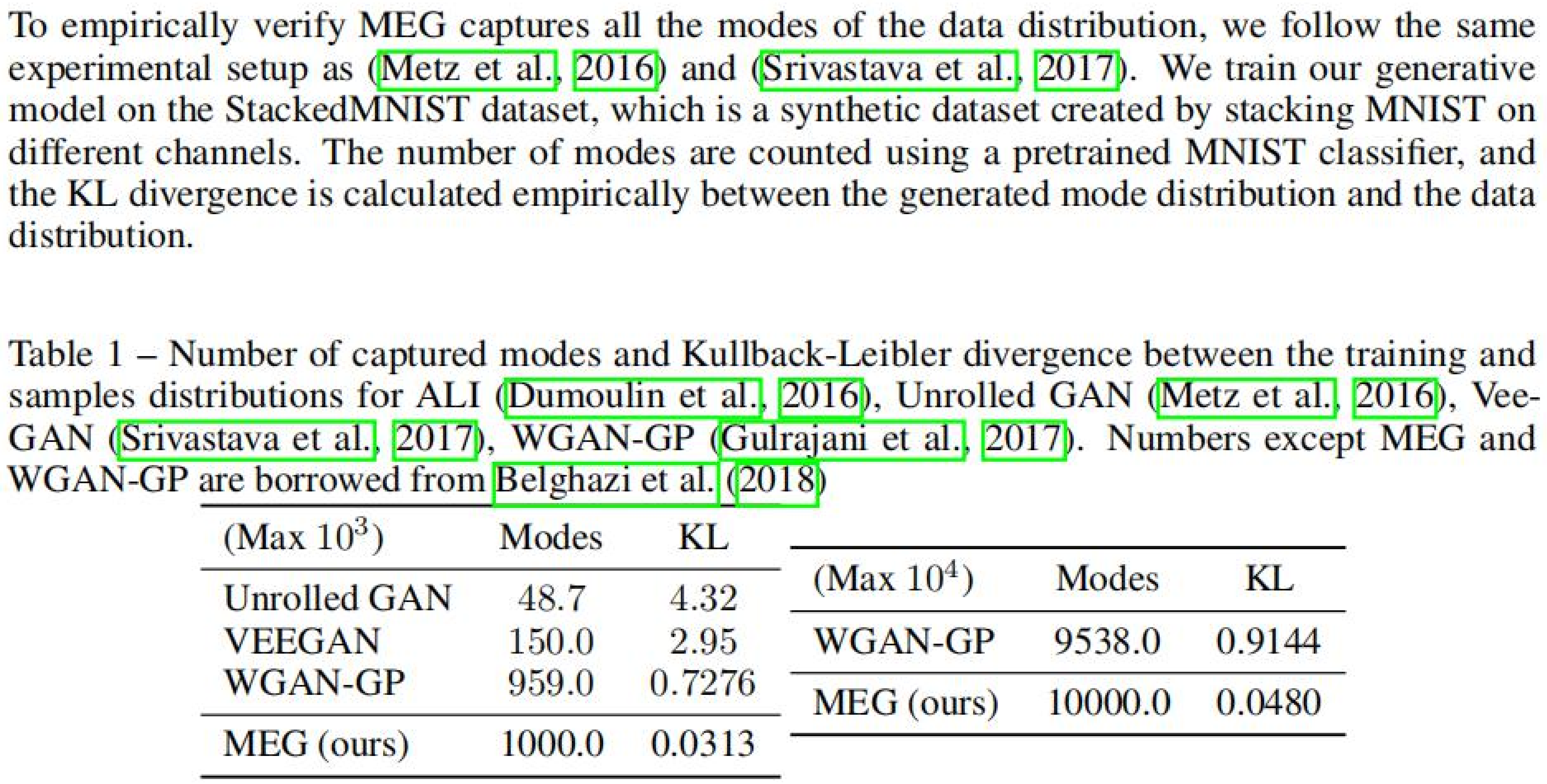


Figure 2 – Probability density visualizations for three popular toy dataset (a) 25-gaussians, (b) swiss roll, and (c) 8-gaussians. Density was estimated using a sample based approximation of the partition function.

Maximum Entropy Generators for EBM



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

Maximum Entropy Generators for EBM

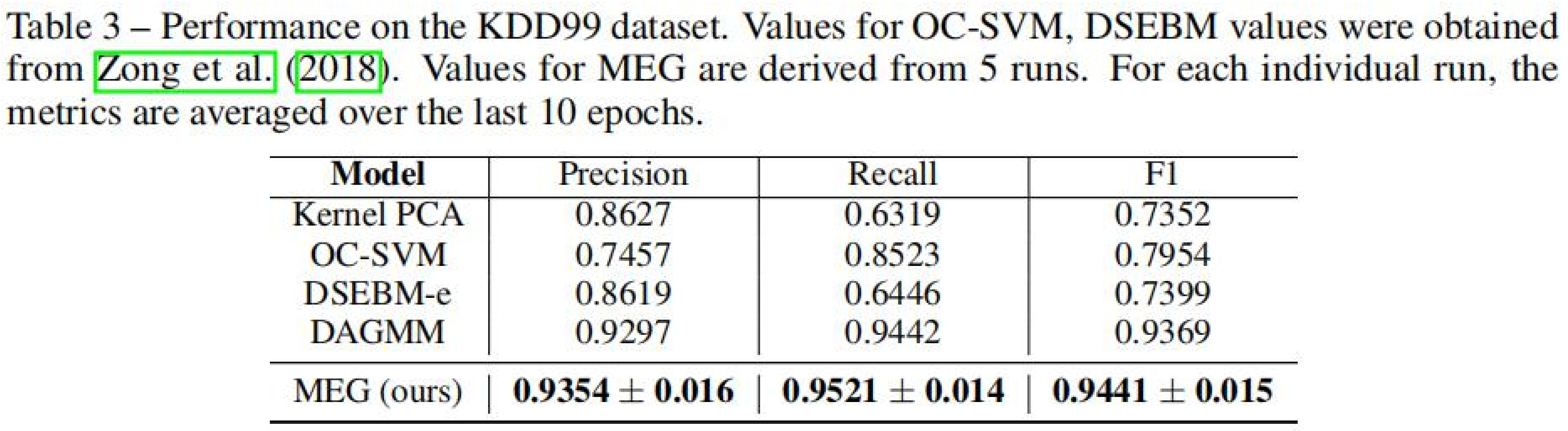


Table 3 – Performance on the KDD99 dataset. Values for OC-SVM, DSEBM values were obtained from Zong et al. (2018). Values for MEG are derived from 5 runs. For each individual run, the metrics are averaged over the last 10 epochs.

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 $\pm$ 0.016**	**0.9521 $\pm$ 0.014**	**0.9441 $\pm$ 0.015**

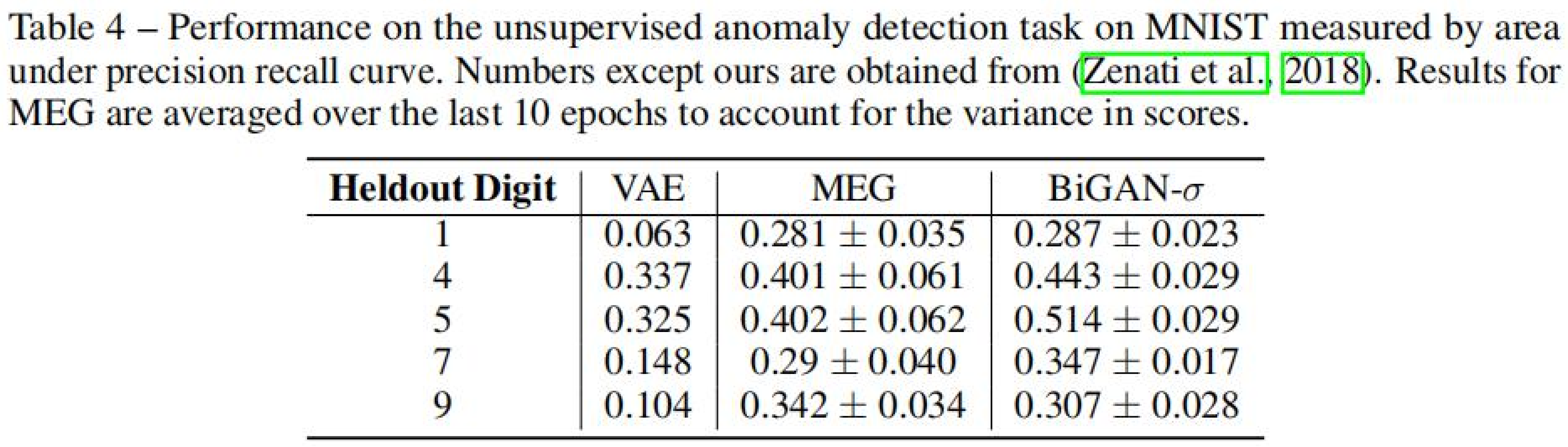
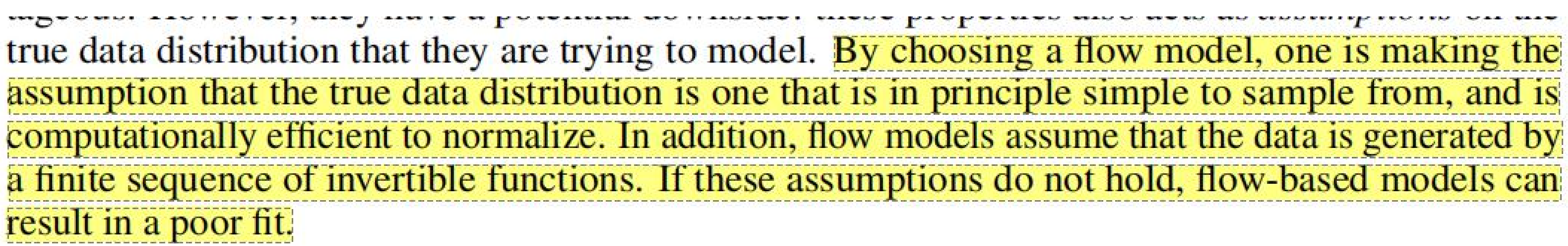


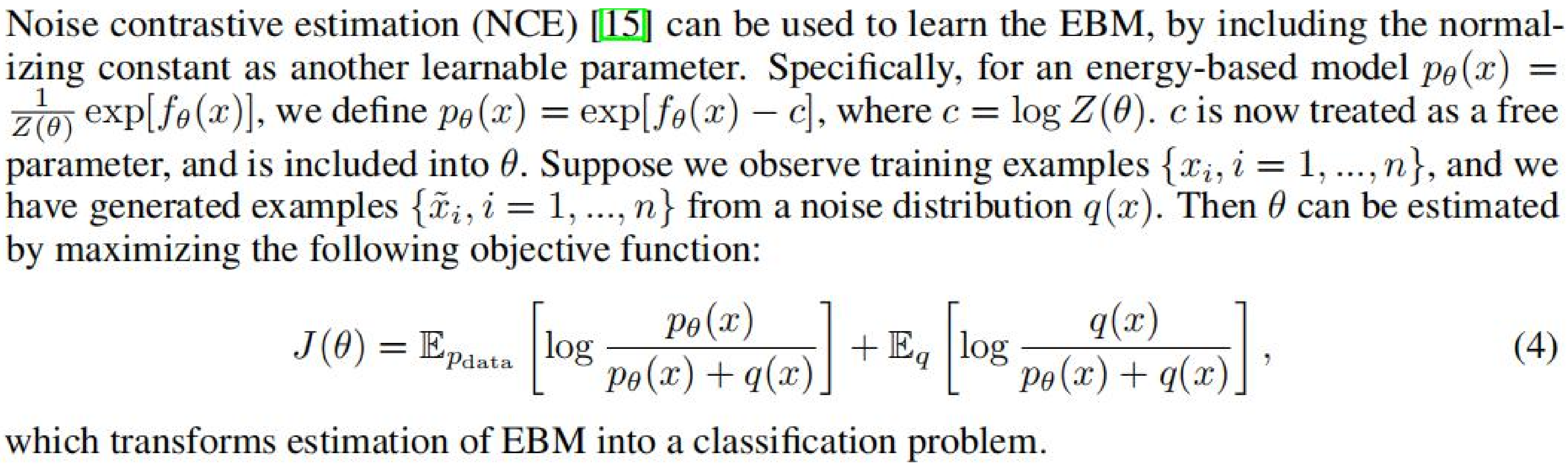
Table 4 – Performance on the unsupervised anomaly detection task on MNIST measured by area under precision recall curve. Numbers except ours are obtained from (Zenati et al., 2018). Results for MEG are averaged over the last 10 epochs to account for the variance in scores.

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

Flow & EBM



true data distribution that they are trying to model. By choosing a flow model, one is making the assumption that the true data distribution is one that is in principle simple to sample from, and is computationally efficient to normalize. In addition, flow models assume that the data is generated by a finite sequence of invertible functions. If these assumptions do not hold, flow-based models can result in a poor fit.



Noise contrastive estimation (NCE) [15] can be used to learn the EBM, by including the normalizing constant as another learnable parameter. Specifically, for an energy-based model $p_\theta(x) = \frac{1}{Z(\theta)} \exp[f_\theta(x)]$, we define $p_\theta(x) = \exp[f_\theta(x) - c]$, where $c = \log Z(\theta)$. c is now treated as a free parameter, and is included into θ. Suppose we observe training examples $\{x_i, i = 1, ..., n\}$, and we have generated examples $\{\tilde{x}_i, i = 1, ..., n\}$ from a noise distribution $q(x)$. Then θ can be estimated by maximizing the following objective function:

$$J(\theta) = \mathbb{E}_{p_{\text{data}}} \left[\log \frac{p_\theta(x)}{p_\theta(x) + q(x)} \right] + \mathbb{E}_q \left[\log \frac{q(x)}{p_\theta(x) + q(x)} \right], \tag{4}$$

which transforms estimation of EBM into a classification problem.

q(x) is a design issue 1. analytically tractable expression of normalized density
2. easy to draw sample from
3. close to data distribution

Flow Contrastive Estimation of Energy-Based Models

3.4 Semi-supervised learning

A class-conditional energy-based model can be transformed into a discriminative model in the following sense. Suppose there are K categories $k = 1, ..., K$, and the model learns a distinct density $p_{\theta_k}(x)$ for each k. The networks $f_{\theta_k}(x)$ for $k = 1, ..., K$ may share common lower layers, but with different top layers. Let ρ_k be the prior probability of category k, for $k = 1, ..., K$. Then the posterior probability for classifying x to the category k is a softmax multi-class classifier

$$P(k|x) = \frac{\exp(f_{\theta_k}(x) + b_k)}{\sum_{l=1}^{K} \exp(f_{\theta_l}(x) + b_l)}, \tag{15}$$

where $b_k = \log(\rho_k) - \log Z(\theta_k)$.

$$\begin{aligned} L_{\text{label}}(\theta) &= \mathbb{E}_{p_{\text{data}}(x,y)} [\log p_\theta(y|x, y \in \{1, ..., K\})] \\ &= \mathbb{E}_{p_{\text{data}}(x,y)} \left[\log \frac{p_{\theta_y}(x)}{\sum_{k=1}^{K} p_{\theta_k}(x)}\right], \end{aligned} \tag{17}$$

which is similar to a classifier in the form.

For unlabeled examples, the probability can be defined by an unconditional EBM, which is in the form of a mixture model:

$$p_\theta(x) = \sum_{i=1}^{K} p_\theta(x|y = k)p(y = k) = \frac{1}{K} \sum_{i=1}^{K} p_{\theta_k}(x), \tag{18}$$

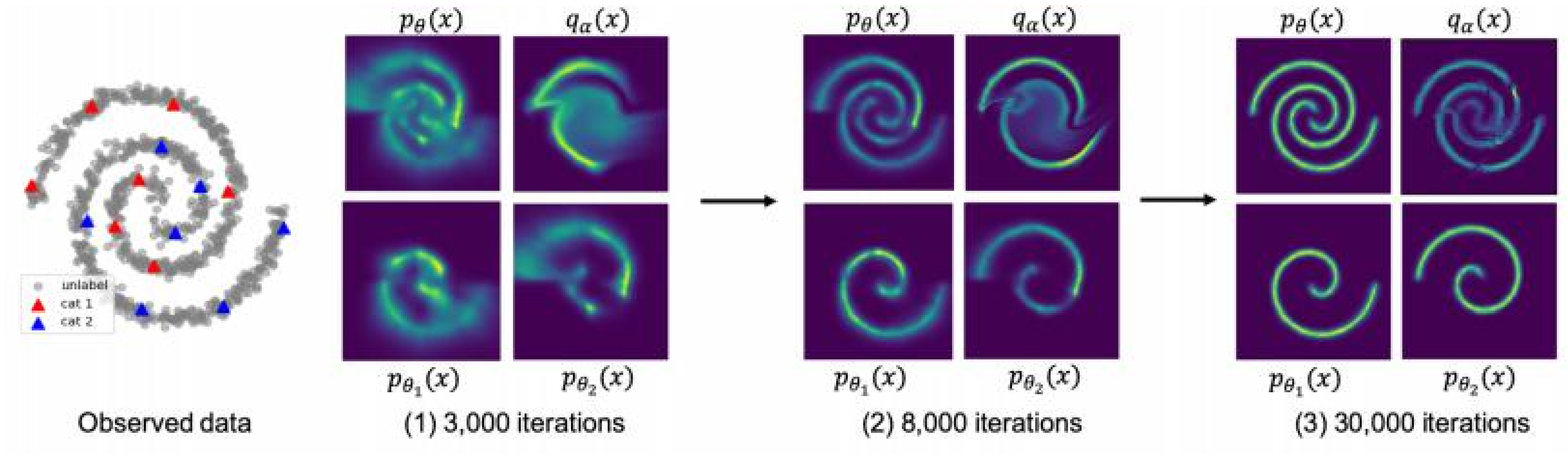


Figure 5: Illustration of FCE for semi-supervised learning on a 2D example, where the data distribution is two spirals belonging to two categories. Within each panel, the top left is the learned unconditional EBM. The top right is the learned Glow model. The bottom are two class-conditional EBMs. For observed data, seven labeled points are provided for each category.

Paper:

Training restricted boltzmann machines using approximations to the likelihood gradient (RBM)

Bayesian learning via stochastic gradient langevin dynamics (langevin dynamics)

Implicit Generation and Modeling with Energy-Based Models

A tutorial on energy-based learning

Learning deep representations by mutual information estimation and maximization (JSD)

Maximum Entropy Generators for Energy-Based Models

Noise-contrastive estimation: A new estimation principle for unnormalized statistical models (NCE)

Flow Contrastive Estimation of Energy-Based Models

Code:

https://github.com/ritheshkumar95/energy_based_generative_models