Flow Contrastive Estimation of Energy-Based Model

Ruiqi Gao
UCLA
ruiqigao@ucla.edu

Erik Nijkamp
UCLA
enijkamp@ucla.edu

Zhen Xu
Google
zhenxu@google.com

Andrew M. Dai
Google
adai@google.com

Diederik P. Kingma
Google
durk@google.com

Ying Nian Wu
UCLA
ywu@stat.ucla.edu

Abstract

This paper studies a training method to jointly learn an energy-based model and a flow-based model, in which the two models are iteratively updated based on a shared adversarial value function. This joint training method has the following traits. (1) The update of the energy-based model is based on noise contrastive estimation, with the flow model serving as the noise distribution. (2) The update of the flow model approximately minimizes the Jensen-Shannon divergence between the flow model and the data distribution. (3) Unlike GAN which learns an implicit probability distribution defined by a generator model, our method learns two explicit probabilistic distributions on the data. We demonstrate the proposed approach on 2D synthetic data and real image datasets.

1 Introduction

Recently flow-based models have gained popularity in generative modeling [3, 4, 19, 10, 1, 23, 40] and variational inference [18, 37, 20, 16, 14]. Unlike the generator model [18, 9] which defines a probability density function implicitly through a low-dimensional hidden vector, a flow model defines a normalized probability density explicitly, making it convenient for various tasks, including learning and sampling. However, in order to achieve tractability, the flow models are forced to compose many layers of transformations of rather unnatural forms. This is also the case with auto-regressive models [35, 42, 38].

Besides flow models, there is yet another class of models that define explicit probability density functions, albeit of unnormalized forms. These are the energy-based models (EBM) [27, 33, 15, 44, 43, 8, 24, 34, 5, 6]. Originated from statistical mechanics, an energy-based model has an unnormalized density that is the exponential of the negative energy function, so that instances of low energies have higher probabilities. Due to the intractability of the normalizing constant, the energy-based model is computationally challenged, since learning and sampling usually require expensive Markov Chain Monte Carlo (MCMC) sampling. However, the energy function can be modeled by a relatively simple network of a natural form, such as a convolutional network [26, 22]. Moreover, the model has a direct correspondence with the commonly used discriminative network, as a simple consequence of the Bayes rule [2, 13, 25], so that the learned network can be turned into a classifier, or the features of the learned network can be used for downstream tasks such as classification.

Contrasting an EBM with a flow model, the former is on the side of representation where different layers represent features of different complexities, whereas the latter is on the side of learned computation, where each layer is like a step in the computation. The EBM is like an objective function, a target distribution, an evaluator or a critic, whereas the flow model is like a finite step iterative algorithm, a sampler or an actor. As a result, the EBM can be simpler and more flexible

4th workshop on Bayesian Deep Learning (NeurIPS 2019), Vancouver, Canada.

in form, and may capture the modes of the data distribution more accurately than the flow model, provided that computation such as sampling is possible. In comparison, the flow model is capable of direct generation via ancestral sampling, which is sorely lacking in EBM. It may thus be desirable to train the two models jointly. This is the goal of this paper.

Our joint training method is inspired by the noise contrastive estimation (NCE) of [11], where an EBM is learned discriminatively by classifying the real data and the data generated by a noise model. See also [41, 13, 25] for a related introspective learning method.

In NCE, the noise model must have an explicit normalized density function. Moreover, it is desirable for the noise distribution to be close to the data distribution for accurate estimation of EBM. However, the noise distribution can be far away from the data distribution. The flow model can potentially transform or transport the noise distribution to a distribution closer to the data distribution. With the advent of strong flow models such as Glow [19], it is natural to recruit the flow model as the contrast distribution for NCE learning of EBM.

However, even with the flow model pre-trained by maximum likelihood estimation (MLE) on the data distribution, it may still not be strong enough as a contrast distribution, in the sense that the synthesized examples generated by the pre-trained flow model may still be distinguished from the real examples by a classifier based on an EBM. Thus, we want the flow model to be a stronger contrast or a stronger training opponent for EBM. To achieve this goal, we can simply use the same objective function of NCE, which is the log-likelihood of the logistic regression for classification. While NCE updates the EBM by maximizing this objective function, we can also update the flow model by minimizing the same objective function to make the classification task harder for EBM. Such update of the flow model combines MLE and variational approximation, and may help correct the over-dispersion of MLE. If the EBM is close to the data distribution, this amounts to minimizing the Jensen-Shannon divergence (JSD) [9] between the data distribution and the flow model. In this sense, the learning scheme relates to GAN closely. However, unlike GAN [9] which learns a generator model that defines an implicit probability density function via a low-dimensional latent vector, our method learns two probabilistic models with explicit probability densities (a normalized one and an unnormalized one).

In the context of inverse reinforcement learning, [28] proposes a guided policy search method, and [6] connects it to GAN. Our method is closely related to this method, where the energy function can be viewed as the cost function, and the flow model can be viewed as the unrolled policy. In our work, we follow NCE learning [11, 30] of EBM, by treating the normalizing constant as a free parameter to be updated together with the original parameters in gradient descent update in discriminative learning, instead of estimating the normalizing constant by importance sampling which may be less reliable in high dimensional situations. Moreover, the flow model is updated by the same objective function as learning the EBM in an adversarial scheme.

The contributions of our paper are as follows. We explore a learning method that couples the learning of EBM and the learning of the flow model using a shared objective function. It improves NCE learning of EBM with a flow-transformed noise distribution, and it modifies the MLE learning of the flow model to approximate JSD learning, which may help correct the over-dispersion of MLE. Experiments on 2D synthetic data show that the learned EBM achieves accurate density estimation with much simpler network structure than the flow model. On real image datasets, the features from the learned EBM are useful for downstream classification task, and the learned flow model achieves high synthesis quality, which is better than the same model trained by MLE.

2 Learning method

2.1 Energy-based model

Let x be the input signal, such as an image. We use $p_\theta(x)$ to denote the probability density function of x with parameter θ. The energy-based model (EBM) is defined as follows [43]:

$$p_\theta(x) = \frac{1}{Z(\theta)} \exp[f_\theta(x)], \tag{1}$$

where f_θ is defined by a bottom-up convolutional neural network whose parameters are denoted by θ. The normalizing constant $Z(\theta) = \int \exp[f_\theta(x)]dx$ is analytically intractable. The model corresponds

to a classifier in the following sense. Suppose there are K categories $p_{\theta_k}(x)$, for $k = 1, ..., K$. The networks $f_{\theta_k}(x)$ for $k = 1, ..., K$ may share common lower layers, but with different heads. 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

$$P(k|x) = \frac{\exp(f_{\theta_k}(x) + b_k)}{\sum_{k=1}^{K} \exp(f_{\theta_k}(x) + b_k)}, \tag{2}$$

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

2.2 Noise contrastive estimation

Noise contrastive estimation (NCE) [11] 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 $c = \log Z(\theta)$, but 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 samples $\{\tilde{x}_i, i = 1, ..., n\}$ from a noise distribution $q(x)$. Then θ can be estimated by maximizing the following log-likelihood for logistic regression:

$$J(\theta) = \sum_{i=1}^{n} \log \left[\frac{p_\theta(x_i)}{p_\theta(x_i) + q(x_i)} \right] + \sum_{i=1}^{n} \log \left[\frac{q(\tilde{x}_i)}{p_\theta(\tilde{x}_i) + q(\tilde{x}_i)} \right], \tag{3}$$

which transforms the learning of EBM into a classification problem. The choice of the noise distribution $q(x)$ is a design issue. We expect $q(x)$ to satisfy the following: (1) analytically tractable expression of normalized density; (2) easy to draw samples from; (3) close to data distribution. In practice, (3) is important for learning high dimensional distribution. If $q(x)$ is not close to the data distribution, the classification problem would be too easy and would not require p_θ to learn much about the structure of the data.

2.3 Joint learning method

A natural improvement to NCE is to transform the noise so that the resulting distribution is closer to the data distribution. This is exactly what the flow model achieves. A flow model is of the form $x = g_\alpha(z)$, where $z \sim q_0(z)$, which is a known noise distribution. g is a composition of a sequence of invertible transformations, and α denotes the parameters. Let $q_\alpha(x)$ be the probability density of x. q_α fulfills (1) and (2) of the requirements of NCE. However, in practice, we find that a pre-trained q_α, such as learned by MLE, is not strong enough for learning EBM p_θ in the sense that the synthesized data from the MLE of q_α can still be easily distinguished from the real data by an EBM. Thus, we propose to iteratively train the EBM and flow model, in which case the flow model is adaptively adjusted to become a stronger contrast distribution or a stronger training opponent for EBM. This is achieved by a learning scheme similar to GAN, where p_θ and q_α play a minimax game with a unified value function: $\min_\alpha \max_\theta V(\theta, \alpha)$,

$$V(\theta, \alpha) = \sum_{i=1}^{n} \log \left[\frac{p_\theta(x_i)}{p_\theta(x_i) + q_\alpha(x_i)} \right] + \mathbb{E}_{z_i, \forall i} \left\{ \sum_{i=1}^{n} \log \left[\frac{q_\alpha(g_\alpha(z_i))}{p_\theta(g_\alpha(z_i)) + q_\alpha(g_\alpha(z_i))} \right] \right\}, \tag{4}$$

where $\{x_i, i = 1, ..., n\}$ are observed samples, and $\{\tilde{x}_i = g_\alpha(z_i), i = 1, ..., n\}$ are negative samples drawn from $q_\alpha(x)$, with $z_i \sim q_0(z)$ independently for $i = 1, ..., n$. We choose Glow [19] as the flow model. The learning can either start from a randomly initialized Glow model or a pre-trained one by MLE. Here we assume equal prior probabilities for observed samples and negative samples. It can be easily modified to the situation where we assign a higher prior probability to the negative samples.

The objective function can be interpreted from the following perspectives:

(1) Noise contrastive estimation for EBM. The update of θ can be seen as noise contrastive estimation of $p_\theta(x)$, but with a flow-transformed noise distribution $q_\alpha(x)$ which is adaptively updated. The training is essentially a logistic regression. However, unlike regular logistic regression for classification, for each x_i or $\tilde{x}_i$, we must include $\log q_\alpha(x_i)$ or $\log q_\alpha(\tilde{x}_i)$ as an example-dependent bias term. This forces $p_\theta(x)$ to replicate $q_\alpha(x)$ in addition to distinguishing between p_{data} and $q_\alpha(x)$, so that $p_\theta(x_i)$ is in general greater than $q_\alpha(x_i)$, and $p_\theta(\tilde{x}_i)$ is in general less than $q_\alpha(\tilde{x}_i)$.

(2) Minimization of Jensen-Shannon divergence for the flow model. If $p_\theta(x)$ is close to the data distribution, then the update of α is approximately minimizing the Jensen-Shannon divergence between the flow model q_α and data distribution p_{data}:

$$\text{JSD}(q_\alpha \| p_{\text{data}}) = \text{KL}(p_{\text{data}} \| (p_{\text{data}} + q_\alpha)/2) + \text{KL}(q_\alpha \| (p_{\text{data}} + q_\alpha)/2). \tag{5}$$

The learning gradient of α follows the gradient of

$$-\mathbb{E}_{p_{\text{data}}}[\log((p_\theta + q_\alpha)/2)] + \text{KL}(q_\alpha \| (p_\theta + q_\alpha)/2). \tag{6}$$

The gradient of the first term resembles MLE learning, which forces q_α to cover the modes of data distribution, which tends to make the learned model over-dispersed. The gradient of the second term is similar to reverse Kullback-Leibler divergence between q_α and p_θ, which forces q_α to chase the modes of p_θ [31, 7]. This may help correct the over-dispersion of MLE.

(3) Connection with GAN. Our learning scheme is similar to GAN. In GAN, the discriminator D and generator G play a minimax game:

$$\min_G \max_D V(G, D) = \sum_{i=1}^{n} \log[D(x_i)] + \sum_{i=1}^{n} \log[1 - D(G(z_i))]. \tag{7}$$

The discriminator $D(x)$ is learning the ratio $p_{\text{data}}(x)/(p_{\text{data}}(x) + p_G(x))$, which is about the difference between p_{data} and p_G [6]. However, in our method, the ratio is explicitly modeled by p_θ and q_α. p_θ must contain all the learned knowledge in q_α, in addition to the difference between p_{data} and q_α. In the end, we learn two explicit probability distributions p_θ and q_α as approximations to p_{data}.

3 Experiments

For joint learning, we adaptively adjust the numbers of updates for EBM and Glow: we first update EBM for a few iterations until the classification accuracy is above 0.5, and then we update Glow until the classification accuracy is below 0.5. We use *Adam* [17] optimizer with 0.001 learning rate for EBM and *Adamax* [17] optimizer with 0.00001 learning rate for Glow. Mini-batch size is 200 for 2D data and 64 for real datasets.

3.1 Density estimation on 2D synthetic data

We first demonstrate our method on 2-dimensional data. Figure 1 shows the result that starts from a randomly intialized Glow. The learned EBM can fit multi-modal distributions accurately, which performs better than Glow either learned by joint learning or by MLE. Notably, the EBM uses a much simpler network structure than Glow: for Glow we use 10 affine coupling layers, which amount to 30 fully-connected layers, while the energy-based model is defined by a 4 layer fully-connected network with the same width as Glow. Another interesting finding is that the EBM can fit the distributions well, even if the flow model is not a perfect contrast distribution.

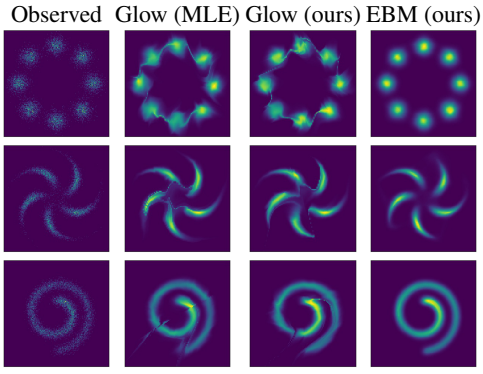

Figure 1: Comparison of trained EBM and Glow models on 2-dimensional distributions.

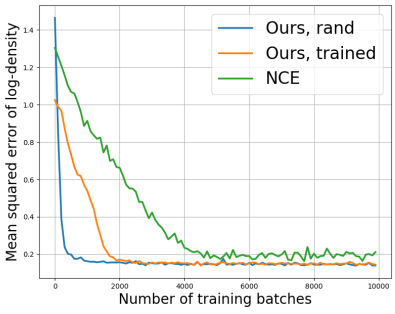

Figure 2: Density estimation accuracy in 2D.

For the mixture of Gaussians, as depicted in the first row of Figure 1, we can compare the estimated density by the learned models with the ground truth. Figure 2 shows the mean squared error of the estimated log-density over numbers of updates of EBM. We show the results for joint learning either starting from a randomly initialized Glow (Ours, rand) or a pre-trained Glow model by MLE (Ours, trained), and compare with the original NCE where the noise distribution is a Gaussian distribution. The learning starting from a randomly initialized Glow converges in less iterations, and both settings of joint learning achieve a lower error rate compared to the original NCE.

3.2 Learning on real image datasets

We conduct experiments on the Street View House Numbers (SVHN) [32] and CIFAR-10 [21] datasets. We start the learning from a pre-trained Glow model by MLE for the sake of efficiency. Again for EBM, we use a simple network with 4 convolutional layers, and we follow the architecture of [19] for Glow model, which has a more complex architecture than EBM. See supplementary A for detailed model architectures. Figure 3 shows synthesized examples from learned Glow models. As shown in Table 1, the generated samples from Glow are improved, in terms of Fréchet Inception Distance (FID) [12] with Inception V3 [39], and also outperform DCGAN [36]. In Table 2, we report the average negative log-likelihood (bits per dimension) on the testing data of SVHN. The log-likelihood of the learned EBM is based on the estimated normalizing constant and should be taken with a grain of salt. The learned flow model has slightly lower log-likelihood (higher bits per dimension) than MLE, but has better synthesis quality.

Figure 3: Synthesized examples from Glow after learning. SVHN on the left, CIFAR-10 on the right.

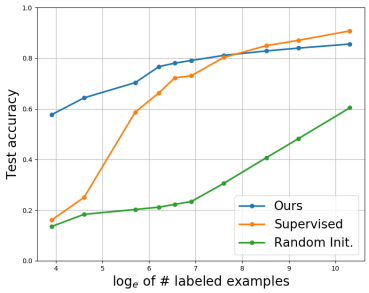

Figure 4: Classification accuracy as a function of number of labeled examples, learned on SVHN.

To further explore if the EBM learns useful features, we extract the top layer feature maps from the EBM learned from SVHN, and train a linear classifier on top of it by a certain amount of labeled images from the training dataset. In Figure 4 we plot the classification accuracy as a function of the number of labeled examples. We compare our method with the supervised classification method trained on the labeled examples. We also show the *Random Init.* setting where a linear classifier is trained on the top layer feature maps of an EBM that is randomly initialized and stays frozen. We observe that our method outperforms the supervised classification method when the number of training images is small.

Table 1: FID score for generated samples

Dataset	DCGAN	Glow (MLE)	Glow (Ours)
SVHN	21.40	41.70	**20.19**
CIFAR-10	37.70	45.99	**37.30**

Table 2: Bits per dimension on testing data

Dataset	Glow (MLE)	Glow (Ours)	EBM (Ours)
SVHN	2.17	2.25	2.15
CIFAR-10	3.35	3.45	3.27

4 Conclusion

In this paper, we study a learning method that couples the learning of an energy-based model with the learning of a flow-based model. The method can be considered an improved version of noise

contrastive estimation where the noise is transformed by a flow model to make its distribution closer to the data distribution and to make it a stronger contrast to the energy-based model.

In our future work, we shall generalize the flow contrastive estimation to K-categories for $K > 2$, and we shall apply the learning method to supervised learning from small labeled data and semi-supervised learning from big unlabeled data and small labeled data. Recently with the generalized method for semi-supervised learning, we achieve results comparable to the start-of-the-art semi-supervised learning methods. Moreover, we intend to generalize the joint learning method by combining the energy-based model with other normalized probabilistic models, such as auto-regressive models.

Acknowledgments

The work is partially supported by DARPA XAI project N66001-17-2-4029. We thank Pavel Sountsov, Alex Alemi and Srinivas Vasudevan for their helpful discussions.

References

[1] Jens Behrmann, David Duvenaud, and Jörn-Henrik Jacobsen. Invertible residual networks. *arXiv preprint arXiv:1811.00995*, 2018.

[2] Jifeng Dai, Yang Lu, and Ying-Nian Wu. Generative modeling of convolutional neural networks. *arXiv preprint arXiv:1412.6296*, 2014.

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

[4] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[5] Yilun Du and Igor Mordatch. Implicit generation and generalization in energy-based models. *arXiv preprint arXiv:1903.08689*, 2019.

[6] Chelsea Finn, Paul Christiano, Pieter Abbeel, and Sergey Levine. A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models. *arXiv preprint arXiv:1611.03852*, 2016.

[7] Charles W Fox and Stephen J Roberts. A tutorial on variational bayesian inference. *Artificial intelligence review*, 38(2):85–95, 2012.

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

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[10] Will Grathwohl, Ricky TQ Chen, Jesse Betterncourt, Ilya Sutskever, and David Duvenaud. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*, 2018.

[11] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304, 2010.

[12] 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.

[13] Long Jin, Justin Lazarow, and Zhuowen Tu. Introspective classification with convolutional nets. In *Advances in Neural Information Processing Systems*, pages 823–833, 2017.

[14] Ilyes Khemakhem, Diederik P Kingma, and Aapo Hyvärinen. Variational autoencoders and nonlinear ica: A unifying framework. *arXiv preprint arXiv:1907.04809*, 2019.

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

[16] Diederik Kingma and Max Welling. Efficient gradient-based inference through transformations between bayes nets and neural nets. In *International Conference on Machine Learning*, pages 1782–1790, 2014.

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[18] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[19] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pages 10215–10224, 2018.

[20] 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.

[21] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[23] Manoj Kumar, Mohammad Babaeizadeh, Dumitru Erhan, Chelsea Finn, Sergey Levine, Laurent Dinh, and Durk Kingma. Videoflow: A flow-based generative model for video. *arXiv preprint arXiv:1903.01434*, 2019.

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

[25] Justin Lazarow, Long Jin, and Zhuowen Tu. Introspective neural networks for generative modeling. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2774–2783, 2017.

[26] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[27] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

[28] Sergey Levine and Vladlen Koltun. Guided policy search. In *International Conference on Machine Learning*, pages 1–9, 2013.

[29] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3, 2013.

[30] Andriy Mnih and Yee Whye Teh. A fast and simple algorithm for training neural probabilistic language models. *arXiv preprint arXiv:1206.6426*, 2012.

[31] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[32] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[33] Jiquan Ngiam, Zhenghao Chen, Pang W Koh, and Andrew Y Ng. Learning deep energy models. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 1105–1112, 2011.

[34] Erik Nijkamp, Song-Chun Zhu, and Ying Nian Wu. On learning non-convergent short-run mcmc toward energy-based model. *arXiv preprint arXiv:1904.09770*, 2019.

[35] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.

[36] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[37] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015.

[38] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017.

[39] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[40] Dustin Tran, Keyon Vafa, Kumar Krishna Agrawal, Laurent Dinh, and Ben Poole. Discrete flows: Invertible generative models of discrete data. *arXiv preprint arXiv:1905.10347*, 2019.

[41] Zhuowen Tu. Learning generative models via discriminative approaches. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[42] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in neural information processing systems*, pages 4790–4798, 2016.

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

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

A Model architectures

For Glow model, we follow the setting of [19]. The architecture has multi-scales with levels L. Within each level, there are K flow blocks. Each block has three convolutional layers (or fully-connected layers) with ReLU activation after the first two layers and a width of W channels. Table 3 summarizes the hyperparameters for different datasets.

Table 3: Hyperparameters for Glow architecture

Dataset	Levels L	Blocks per level K	Width W	Layer type	Coupling
2D data	1	10	128	fc	affine
SVHN	3	8	512	conv	additive
CIFAR-10	3	32	512	conv	additive

Table 4 summarizes the EBM architecture. The slope of all leaky ReLU (lReLU) [29] functions are set to 0.2.

Table 4: EBM architectures

2D data	SVHN / CIFAR-10
fc. 128 lReLU	4×4 conv. 64 lReLU, stride 2
fc. 128 lReLU	4×4 conv. 128 lReLU, stride 2
fc. 128 lReLU	4×4 conv. 256 lReLU, stride 2
fc. 1	4×4 conv. 1, stride 1

B Synthesis comparison

In Figures 5 and 6, we display the synthesized examples from Glow trained by MLE and joint learning.

Figure 5: Synthesis examples from Glow learned from SVHN. Left panel is by MLE. Right panel is by our joint learning.

Figure 6: Synthesis examples from Glow learned from CIFAR-10. Left panel is by MLE. Right panel is by our joint learning.