ENERGY-BASED GENERATIVE ADVERSARIAL NETWORKS

Junbo Zhao, Michael Mathieu and Yann LeCun
Department of Computer Science, New York University
Facebook Artificial Intelligence Research
{jakezhao, mathieu, yann}@cs.nyu.edu

ABSTRACT

We introduce the "Energy-based Generative Adversarial Network" model (EBGAN) which views the discriminator as an energy function that attributes low energies to the regions near the data manifold and higher energies to other regions. Similar to the probabilistic GANs, a generator is seen as being trained to produce contrastive samples with minimal energies, while the discriminator is trained to assign high energies to these generated samples. Viewing the discriminator as an energy function allows to use a wide variety of architectures and loss functionals in addition to the usual binary classifier with logistic output. Among them, we show one instantiation of EBGAN framework as using an auto-encoder architecture, with the energy being the reconstruction error, in place of the discriminator. We show that this form of EBGAN exhibits more stable behavior than regular GANs during training. We also show that a single-scale architecture can be trained to generate high-resolution images.

1 INTRODUCTION

1.1 ENERGY-BASED MODEL

The essence of the energy-based model (LeCun et al., 2006) is to build a function that maps each point of an input space to a single scalar, which is called "energy". The learning phase is a data-driven process that shapes the energy surface in such a way that the desired configuration gets assigned low energies, while the incorrect ones are given high energies. Supervised learning falls into this framework: for each X in the training set, the energy of the pair (X, Y) takes low values when Y is the correct label and higher values for incorrect Y's. Similarly, when modeling X alone within an unsupervised learning setting, lower energy is attributed to the data manifold. The term *contrastive sample* is often used to refer to a data point causing an energy pull-up, such as the incorrect Y's in supervised learning and points from low data density regions in unsupervised learning.

1.2 GENERATIVE ADVERSARIAL NETWORKS

Generative Adversarial Networks (GAN) (Goodfellow et al., 2014) have led to significant improvements in image generation (Denton et al., 2015; Radford et al., 2015; Im et al., 2016; Salimans et al., 2016), video prediction (Mathieu et al., 2015) and a number of other domains. The basic idea of GAN is to simultaneously train a discriminator and a generator. The discriminator is trained to distinguish *real* samples of a dataset from *fake* samples produced by the generator. The generator uses input from an easy-to-sample random source, and is trained to produce fake samples that the discriminator cannot distinguish from real data samples. During training, the generator receives the gradient of the output of the discriminator with respect to the fake sample. In the original formulation of GAN in Goodfellow et al. (2014), the discriminator produces a probability and, under certain conditions, convergence occurs when the distribution produced by the generator matches the data distribution. From a game theory point of view, the convergence of a GAN is reached when the generator and the discriminator reach a Nash equilibrium.

1.3 ENERGY-BASED GENERATIVE ADVERSARIAL NETWORKS

In this work, we propose to view the discriminator as an energy function (or a contrast function) without explicit probabilistic interpretation. The energy function computed by the discriminator can be viewed as a trainable cost function for the generator. The discriminator is trained to assign low energy values to regions of high data density, and higher energy values outside the regions of high data density. Conversely, the generator can be viewed as a trainable parameterized function that produces samples in regions of the space to which the generator assigns low energy. While it is often possible to convert energies into probabilities through a Gibbs distribution (LeCun et al., 2006), the absence of normalization in this energy-based form of GAN provides greater flexibility in the choice of architectures of the discriminator and the training procedures.

The probabilistic binary discriminator in the original formulation of GAN can be seen as one way among many to define the contrast function and loss functional, as described in LeCun et al. (2006) for the supervised and weakly supervised settings, and Ranzato et al. (2007) for the unsupervised setting. We experimentally demonstrate this concept, in the setting where the discriminator is an auto-encoder architecture, and the energy is the reconstruction error. More details of the formulation of EBGAN are provided in the appendix B.

Our main contributions are summarized as follows:

- An energy-based formulation for generative adversarial training.
- A proof that under a simple hinge loss, when the system reaches convergence, the generator of EBGAN produces points that follow the true dataset distribution.
- An EBGAN framework with the discriminator using an auto-encoder architecture in which the energy is the reconstruction error.
- A set of systematic experiments to explore the set of hyper-parameters and architectural choices that produce good results for EBGANs and conventional GANs. These experiments demonstrate EBGAN framework to be more robust with respect to the choice of hyper-parameter and architecture.
- A demonstration that EBGANs can generate reasonable-looking high-resolution images from the ImageNet dataset at 256×256 pixel resolution, without a multi-scale approach.

2 THE EBGAN MODEL

Let p_{data} be the underlying probability density of the distribution that produces the dataset. The generator G is trained to produce a sample $G(z)$, for instance an image, from a random vector z, which is sampled from a known distribution p_z, for instance $\mathcal{N}(0, 1)$. The discriminator D takes either real or generated images, and estimates the energy value $E \in \mathbb{R}$ accordingly, as explained later. For simplicity, we assume that D produces non-negative values, but the analysis would hold as long as the values are bounded below.

2.1 OBJECTIVE FUNCTIONAL

The output of the discriminator goes through an objective functional in order to shape the energy function, attributing low energy to the real data samples and higher energy to the generated ("fake") ones. In this work, we use a margin loss, but many other choices are possible in LeCun et al. (2006). Similarly to what has been done with the "classical" GAN (Goodfellow et al., 2014), in order to get better quality gradients when the generator is far from convergence, we use a two different losses, one to train D and the other to train G.
Given a positive margin m, a data sample x and a generated sample $G(z)$, the discriminator loss $\mathcal{L}_D$ and the generator loss $\mathcal{L}_G$ are formally defined:

$$\mathcal{L}_D(x, z) = D(x) + [m - D\big(G(z)\big)]^+ \tag{1}$$

$$\mathcal{L}_G(z) = D\big(G(z)\big) \tag{2}$$

where $[\cdot]^+ = max(0, \cdot)$. Minimizing $\mathcal{L}_G$ with respect to the parameters of G is similar to maximizing the second term of $\mathcal{L}_D$. It has the same minimum but non-zero gradients when $D(G(z)) \geq m$.

2.2 OPTIMALITY OF THE SOLUTION

In this section, we present a theoretical analysis of the system presented in section 2.1. We show that if the system reaches a Nash equilibrium, then the generator G produces samples that are indistinguishable from the distribution of the dataset. This section is done in a non-parametric setting, *i.e.* we assume that D and G have infinite capacity.

Given a generator G, let p_G be the density distribution of $G(z)$ where $z \sim p_z$. In other words, p_G is the density distribution of the samples generated by G.
We define $V(G,D) = \int_{x,z} \mathcal{L}_D(x,z) p_{data}(x) p_z(z) \mathrm{d}x \mathrm{d}z$ and $U(G,D) = \int_z \mathcal{L}_G(z) p_z(z) \mathrm{d}z$. We train the discriminator D to minimize the quantity V and the generator G to minimize the quantity U.
A Nash equilibrium of the system is a pair (G^*, D^*) that satisfies:

$$V(G^*, D^*) \leq V(G^*, D) \qquad \forall D \tag{3}$$

$$U(G^*, D^*) \leq U(G, D^*) \qquad \forall G \tag{4}$$

Theorem 1. *If (D^*, G^*) is a Nash equilibrium of the system, then $p_{G^*} = p_{data}$ almost everywhere, and $V(D^*, G^*) = m$.*

Proof. First we observe that

$$V(G^*, D) = \int_x p_{data}(x) D(x) \mathrm{d}x + \int_z p_z(z) \left[m - D(G^*(z))\right]^+ \mathrm{d}z \tag{5}$$

$$= \int_x \left(p_{data}(x) D(x) + p_{G^*}(x) \left[m - D(x)\right]^+ \right) \mathrm{d}x. \tag{6}$$

The analysis of the function $\varphi(y) = ay + b(m-y)^+$ (see lemma 1 in appendix A for details) shows:
(a) $D^*(x) \leq m$ almost everywhere. To verify it, let us assume that there exists a set of measure non-zero such that $D^*(x) > m$. Let $D'(x) = D^*(x)$ if $D^*(x) \leq m$ and $D'(x) = m$ otherwise. Then $V(G^*, D') < V(G^*, D^*)$ which violates equation 3.
(b) The function φ reaches its minimum in m if $a < b$ and in 0 otherwise. So $V(G^*, D)$ reaches its minimum when we replace $D^*(x)$ by these values. We obtain

$$V(G^*, D^*) = m \int_x \mathbb{1}_{p_{data}(x) < p_{G^*}(x)} p_{data}(x) \mathrm{d}x + m \int_x \mathbb{1}_{p_{data}(x) \geq p_{G^*}(x)} p_{G^*}(x) \mathrm{d}x \tag{7}$$

$$= m \int_x \left(\mathbb{1}_{p_{data}(x) < p_{G^*}(x)} p_{data}(x) + \left(1 - \mathbb{1}_{p_{data}(x) < p_{G^*}(x)}\right) p_{G^*}(x) \right) \mathrm{d}x \tag{8}$$

$$= m \int_x p_{G^*}(x) \mathrm{d}x + m \int_x \mathbb{1}_{p_{data}(x) < p_{G^*}(x)} (p_{data}(x) - p_{G^*}(x)) \mathrm{d}x \tag{9}$$

$$= m + m \int_x \mathbb{1}_{p_{data}(x) < p_{G^*}(x)} (p_{data}(x) - p_{G^*}(x)) \mathrm{d}x. \tag{10}$$

The second term in equation 10 is non-positive, so $V(G^*, D^*) \leq m$.

By putting the ideal generator that generates p_{data} into the right side of equation 4, we get

$$\int_x p_{G^*}(x) D^*(x) \mathrm{d}x \leq \int_x p_{data}(x) D^*(x) \mathrm{d}x. \tag{11}$$

$$\text{Thus by (6),} \qquad \int_x p_{G^*}(x) D^*(x) \mathrm{d}x + \int_x p_{G^*}(x) [m - D^*(x)]^+ \mathrm{d}x \leq V(G^*, D^*) \tag{12}$$

and since $D^*(x) \leq m$, we get $m \leq V(G^*, D^*)$.

Thus, $m \leq V(G^*, D^*) \leq m$ *i.e.* $V(G^*, D^*) = m$. Using equation 10, we see that can only happen if $\int_x \mathbb{1}_{p_{data}(x) < p_G(x)} \mathrm{d}x = 0$, which is true if and only if $p_G = p_{data}$ almost everywhere (this is because p_{data} and p_G are probabilities densities, see lemma 2 in the appendix A for details). □

Theorem 2. *Nash equilibrium of this system exists and is characterized by (a) $p_{G^*} = p_{data}$ (almost everywhere) and (b) there exists a constant $\gamma \in [0, m]$ such that $D^*(x) = \gamma$ (almost everywhere).*[1].

Proof. See appendix A. □

[1]This is assuming there is no region where $p_{data}(x) = 0$. If such a region exists, $D^*(x)$ may have any value in $[0, m]$ for x in this region.

2.3 USING AUTO-ENCODERS

In our experiments, the discriminator D is structured as an auto-encoder:

$$D(x) = ||Dec(Enc(x)) - x||. \tag{13}$$

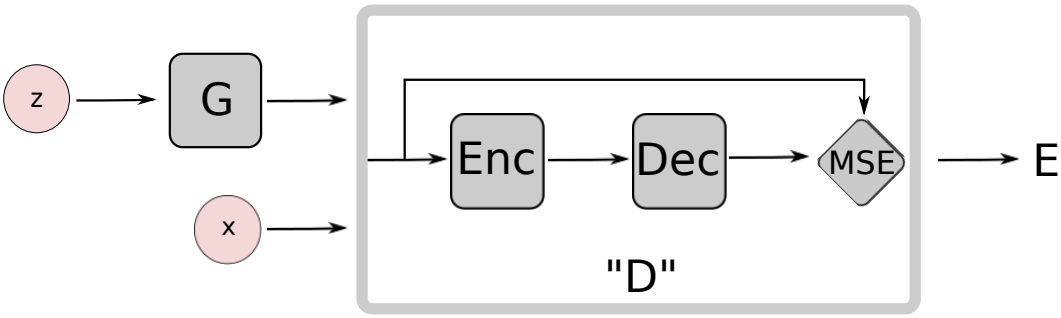

Figure 1: EBGAN architecture with an auto-encoder discriminator.

The diagram of the EBGAN model with an auto-encoder discriminator is depicted in figure 1. The choice of the auto-encoders for D may seem arbitrary at the first glance, yet we postulate that it is conceptually more attractive than a binary logistic network:

- Rather than using a single bit of target information to train the model, the reconstruction-based output offers a diverse targets for the discriminator. With the binary logistic loss, only two targets are possible, so within a minibatch, the gradients corresponding to different samples are most likely far from orthogonal. This leads to inefficient training, and reducing the minibatch sizes is often not an option on current hardware. On the other hand, the reconstruction loss will likely produce very different gradient directions within the minibatch, allowing for larger minibatch size without loss of efficiency.
- Auto-encoders have traditionally been used to represent energy-based model and arise naturally. Given some regularization (see section 2.3.1), auto-encoders have the ability to learn an energy manifold without supervision or negative examples. This mean that even when an EBGAN auto-encoding model is trained to reconstruct a *real* sample, the discriminator contributes to discovering the data manifold by itself. To the contrary, without the presence of negative examples from the generator, a discriminator trained with binary logistic loss becomes pointless.

2.3.1 CONNECTION TO THE REGULARIZED AUTO-ENCODERS

One common issue in training auto-encoders is that the model may learn little more than an identity function. From an energy-based perspective, this means attributing low energy to the whole space. In order to avoid this problem, the model must be pushed to give higher energy to points outside of the data manifold. Theoretical and experimental results have addressed this issue by regularizing the latent representations (Vincent et al., 2010; Rifai et al., 2011; MarcAurelio Ranzato & Chopra, 2007; Kavukcuoglu et al., 2010). Such regularizers aim at restricting the reconstructing power of the auto-encoder so that it can only attribute low energy to a smaller portion of the input points.

We argue that the energy function (the discriminator) in the EBGAN framework can also be seen as "regularized" by having a generator producing the contrastive samples, to which the discriminator ought to give high reconstruction energies. We further argue that the EBGAN framework allows more flexibility from this perspective, because: (i)-the regularizer (generator) is fully trainable instead of being handcrafted; (ii)-the adversarial training paradigm enables a direct interaction between the processes of producing contrastive samples and learning the energy function.

Furthermore, recent work such as Larsen et al. (2015) addresses the insufficient capacity of the ℓ_2 loss function; the authors show that training a variational auto-encoder with the element-wise ℓ_2 loss failed to capture the fine details in its reconstruction. However, we argue that the EBGAN framework is established from an orthogonal angle where the ℓ_2 loss function (or any element-wise loss we may choose) merely serves to produce an energy. It is not a problem if the discriminator of an EBGAN does not reconstructs perfectly, as long as the it is able to tell apart real and fake images by the energies. It is the generator which is producing the final samples.

2.4 REPELLING REGULARIZER

We propose a "repelling regularizer" which fits well into the EBGAN auto-encoder model, to keep the model from producing samples that are clustered in one or a few modes of p_{data}. Another technique "minibatch discrimination" was developed in Salimans et al. (2016) from the same philosophy.

Implementing repelling regularizer has a pulling-away (PT) effect at a representation level. Formally, let $S \in \mathbb{R}^{s \times N}$ denotes a batch of sample representations taken from the encoder output layer. The PT term is defines as:

$$f_{PT}(S) = \frac{1}{N(N-1)} \sum_i \sum_{j \neq i} \Big(\frac{S_i^\intercal S_j}{\|S_i\| \|S_j\|} \Big)^2. \tag{14}$$

The PT term is intended to decrease the magnitude of cosine similarity between pairwise sample representations, and thus making them as orthogonal as possible. Prior work showed that the output layer of encoder carries representational powerful information for various tasks (Rasmus et al., 2015; Zhao et al., 2015). The rationale for choosing the cosine similarity instead of Euclidean distance is to make the term bounded below and invariant to scale. We use the notation "EBGAN-PT" to refer to the EBGAN auto-encoder model trained with this PT term. Note the PT term is used in the generator loss but not in the discriminator loss, where a weight of 0.1 is associated with it when being added to the loss.

3 RELATED WORK

Our work primarily casts GANs into an energy-based model scope. Besides the various type of regularized auto-encoders that the EBGAN framework is connected to (see section 2.3.1), the approaches producing contrastive samples are also highly relevant, such as the use of noisy samples (Vincent et al., 2010) and noisy gradient descent methods including contrastive divergence (Carreira-Perpinan & Hinton, 2005). Several papers was presented with helpful techniques for stabilizing GAN training, (Salimans et al., 2016; Denton et al., 2015; Radford et al., 2015; Im et al., 2016; Mathieu et al., 2015). To our knowledge, our approach is novel and being developed on a model level.

4 EXPERIMENTS

4.1 EXHAUSTIVE GRID SEARCH ON MNIST

In this section we demonstrate the better training stability of EBGANs over GANs on the simple task of MNIST digit generation with fully-connected networks. We run an exhaustive grid search over a set of architectural choices and hyper-parameters for both frameworks. The convolutional architectures applied on larger scale and more complex datasets are exhibited in later sections.

Formally, we specify the search grid in table 1. We impose the following restrictions on EBGANs: (i)-using learning rate 0.001 and Adam (Kingma & Ba, 2014) for both G and D; (ii)-`nLayerD` represents the total number of layers of Enc and Dec put together. Also for simplicity, only the number of layers of Enc is varied, Dec is always a single layer; (iii)-the margin is set to 10 throughout all experiments of EBGANs. To analyze the results of the large grid search, we use the *inception score* (Salimans et al., 2016) as a numerical means for assessing generation quality . We further tweak it and define it as $I' = E_x KL(p(y)||p(y|\mathbf{x}))^2$ to plot more compact histograms (more details see appendix C). A higher inception score corresponds to better quality of the generated images.

Histograms We plot the histogram of I' scores in figure 2. We further separated out the optimization related setting from GAN's grid (`optimD`, `optimG` and `lr`) and plot the histogram of each subgrid individually, together with the EBGAN scores as a reference, in figure 3. The number of experiments for GANs and EBGANs are both 512 in every subplot. The histograms are intended to show that EBGANs are generally more reliably trained than GANs. GANs conversely may demand exquisitely tuned architectural setting and hyper-parameter.

Digits generated with the configurations presenting the best inception score are shown in figure 4.

[2]Although the tweaked inception score should convey the exact information as its original form, it is only used to better analyze the grid search in this work, but not to compare with other published works.

Table 1: Grid search specs

Settings	Description	EBGANs	GANs
`nLayerG`	number of layers in G	[2, 3, 4, 5]	[2, 3, 4, 5]
`nLayerD`	number of layers in D	[2, 3, 4, 5]	[2, 3, 4, 5]
`sizeG`	number of neurons in G	[400, 800, 1600, 3200]	[400, 800, 1600, 3200]
`sizeD`	number of neurons in D	[128, 256, 512, 1024]	[128, 256, 512, 1024]
`dropoutD`	if to use dropout in D	[true, false]	[true, false]
`optimD`	to use Adam or SGD for D	adam	[adam, sgd]
`optimG`	to use Adam or SGD for G	adam	[adam, sgd]
`lr`	learning rate	0.001	[0.01, 0.001, 0.0001]
#experiments:	-	**512**	**6144**

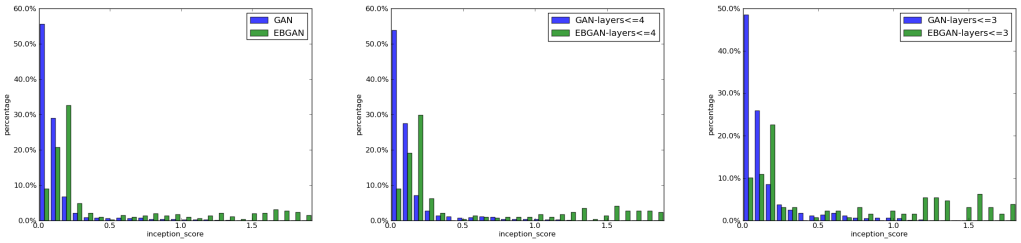

Figure 2: **(Zooming in on pdf file is recommended.)** Histogram of the inception scores from the grid search. The x-axis carries the inception score I and y-axis informs the portion of the models (in percentage) falling into certain bins. Left (a): general comparison of EBGANs against GANs; Middle (b): EBGANs and GANs both constrained by `nLayer[GD]<=4`; Right (c): EBGANs and GANs both constrained by `nLayer[GD]<=3`.

4.2 Semi-supervised learning on MNIST

We examine the possibility of using the EBGAN framework for semi-supervised learning on permutation-invariant MNIST, respectively with 100, 200 and 1000 labels. We utilized a bottom-layer-cost ladder network (LN) (Rasmus et al., 2015) with the EGBAN framework. Ladder Networks can be categorized as an energy-based model that is built with both feedforward and feedback hierarchies with powerful lateral connections coupling two pathways. From table 2, it shows that positioning a bottom-layer-cost LN into an EBGAN framework profitably improves the performance of the LN itself. Albeit slightly lower than the state-of-the-art result by Salimans et al. (2016), our result does shed some light on the postulation that within the scope of the EBGAN framework, iteratively feeding the adversarial contrastive samples produced by the generator to the energy function acts to be an effective regularizer. We notice there was a discrepancy between the reported results between Rasmus et al. (2015) and Pezeshki et al. (2015), so we report both results along with our own implementation of the Ladder Network, which is obtained by running the same setting. The specific experimental setting and analysis is available in appendix D.

Table 2: The comparison of LN bottom-layer-cost model and its EBGAN extension on PI-MNIST semi-supervised task. Note the results are error rate (in %) and they were averaged over 15 different random seeds.

model	100	200	1000
LN bottom-layer-cost, reported in Pezeshki et al. (2015)	1.69±0.18	-	1.05±0.02
LN bottom-layer-cost, reported in Rasmus et al. (2015)	1.09±0.32	-	0.90±0.05
LN bottom-layer-cost, reproduced in this work (see appendix D)	1.36±0.21	1.24±0.09	1.04±0.06
LN bottom-layer-cost within EBGAN framework	**1.04±0.12**	**0.99±0.12**	**0.89±0.04**
Relative percentage improvement	23.5%	20.2%	14.4%

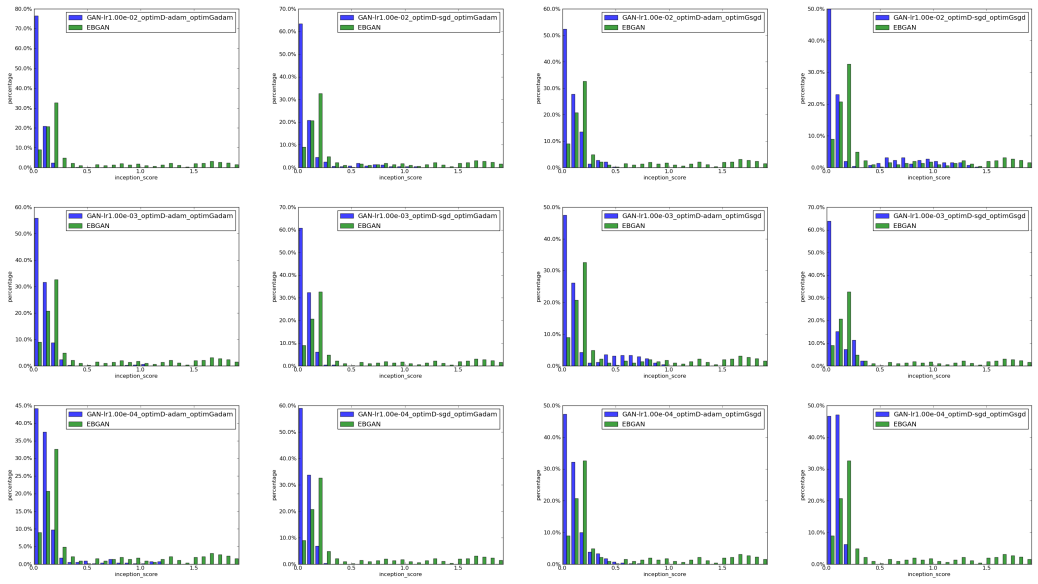

Figure 3: **(Zooming in on pdf file is recommended.)** Histogram of the inception scores grouped by different optimization combinations, drawn from `optimD`, `optimG` and `lr` (See text).

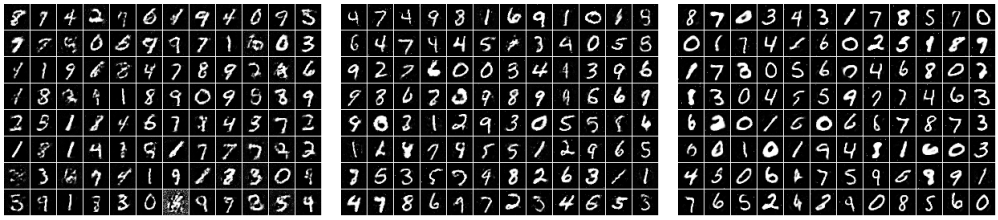

Figure 4: Generation from the grid search on MNIST. Left(a): Best GAN model; Middle(b): Best EBGAN model. Right(c): Best EBGAN-PT model.

4.3 LSUN & CELEBA

We apply the EBGAN framework with deep convolutional architecture to generate 64×64 RGB images, a more realistic task, using the LSUN bedroom dataset (Yu et al., 2015) and the large-scale face dataset CelebA under alignment (Liu et al., 2015). To compare EBGANs with DCGANs (Radford et al., 2015), we train a DCGAN model under the same configuration and show the generations side-by-side with the EBGAN model, in figures 5 and 6. The specific settings are listed in appendix C.

4.4 IMAGENET

Finally, we trained EBGANs to generate high-resolution images on ImageNet (Russakovsky et al., 2015). Compared with the datasets we have experimented so far, ImageNet presents an extensively larger and wilder space, so modeling the data distribution in a generative model becomes very challenging. We devised an experiment to generate 128×128 images, trained on the full ImageNet-1k dataset, which contains roughly 1.3 million images from 1000 different categories. We also trained a network to generate images of size 256×256, on a dog-breed subset of ImageNet, using the wordNet IDs provided by Vinyals et al. (2016). The results are shown in figures 7 and 8. Despite the difficulty of generating images on a high-resolution level, we observe that EBGANs are able to learn about the fact that objects appear in the foreground, together with various background components resembling grass texture, sea under the horizon, mirrored mountain in the water, buildings, etc. In addition, our 256×256 dog-breed generations, although far from realistic, show knowledge about the appearances of dogs such as their body, furs and eye.

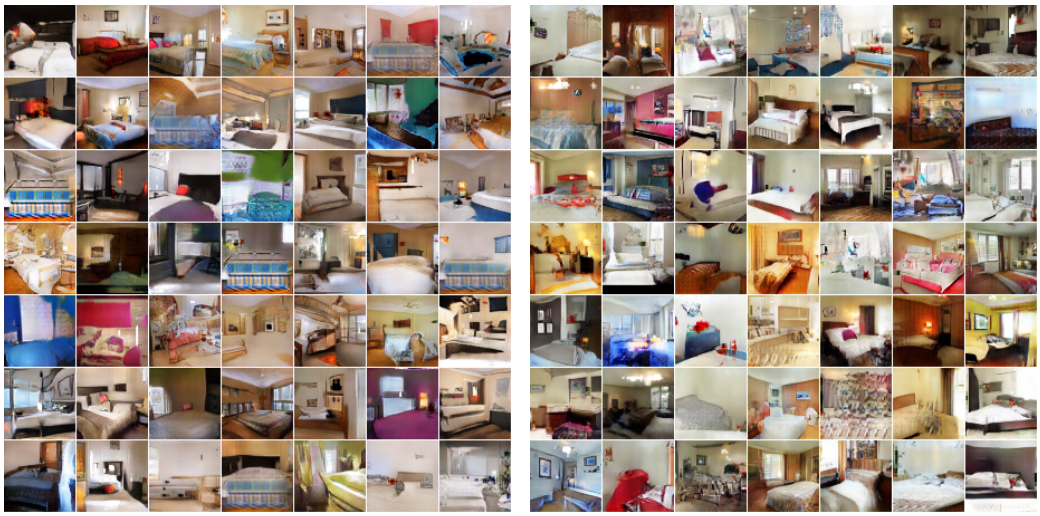

Figure 5: Generation from LSUN bedroom full-images. Left(a): DCGAN generation. Right(b): EBGAN-PT generation.

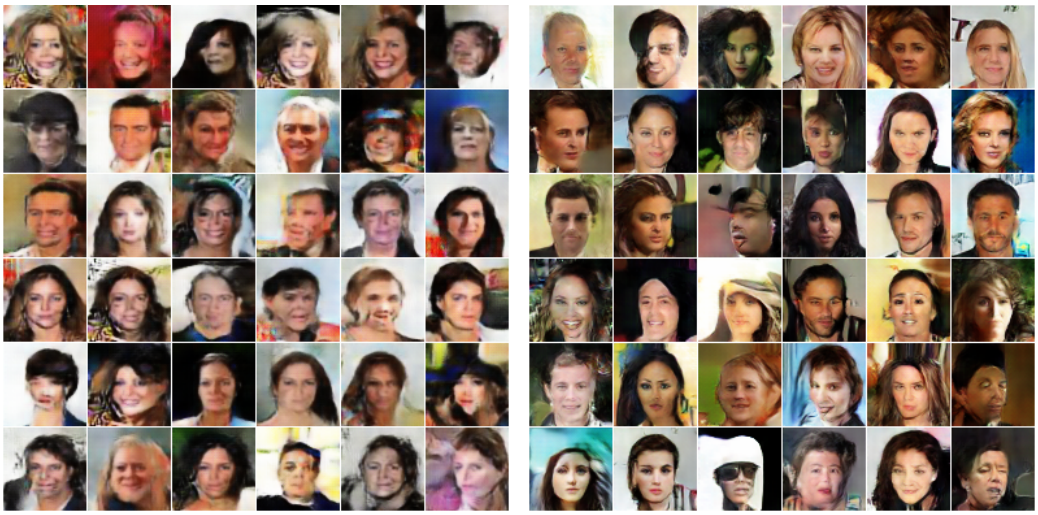

Figure 6: Generation from CelebA face dataset. Left(a): DCGAN generation. Right(b): EBGAN-PT generation.

5 OUTLOOK

We bridge two classes of unsupervised learning methods – GANs and auto-encoders – and revisit the GAN framework from an alternative energy-based perspective. EBGANs show better convergence pattern and scalability to generate high-resolution images. A family of energy-based loss functionals presented in LeCun et al. (2006) can easily be incorporated into the EBGAN framework. For the future work, the conditional setting (Denton et al., 2015; Mathieu et al., 2015) is a promising setup to explore. We hope the future research will raise more attention on a broader view of GANs from the energy-based perspective.

ACKNOWLEDGMENT

We thank Emily Denton, Soumith Chitala, Arthur Szlam, Marc'Aurelio Ranzato, Pablo Sprechmann, Ross Goroshin and Ruoyu Sun for fruitful discussions. We also thank Emily Denton and Tian Jiang for their help with the manuscript.

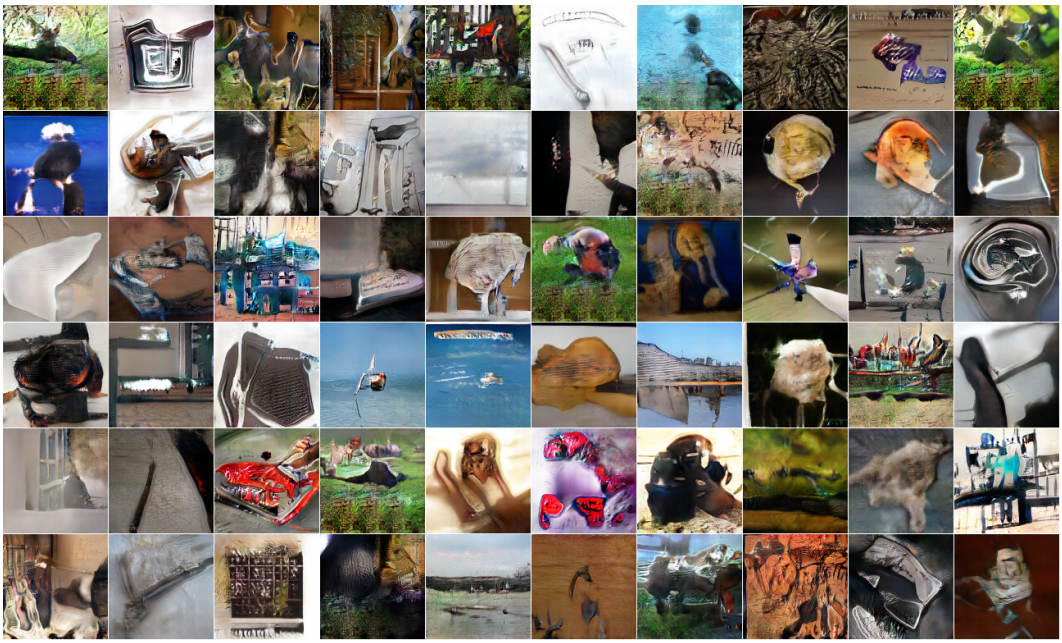

Figure 7: ImageNet 128×128 generations using an EBGAN-PT.

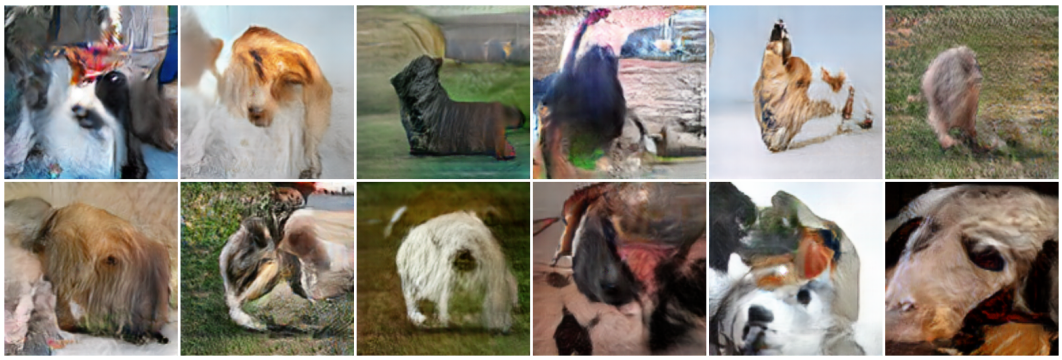

Figure 8: ImageNet 256×256 generations using an EBGAN-PT.

REFERENCES

Carreira-Perpinan, Miguel A and Hinton, Geoffrey. On contrastive divergence learning. In *AISTATS*, volume 10, pp. 33–40. Citeseer, 2005.

Denton, Emily L, Chintala, Soumith, Fergus, Rob, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems*, pp. 1486–1494, 2015.

Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

Im, Daniel Jiwoong, Kim, Chris Dongjoo, Jiang, Hui, and Memisevic, Roland. Generating images with recurrent adversarial networks. *arXiv preprint arXiv:1602.05110*, 2016.

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

Kavukcuoglu, Koray, Sermanet, Pierre, Boureau, Y-Lan, Gregor, Karol, Mathieu, Michaël, and Cun, Yann L. Learning convolutional feature hierarchies for visual recognition. In *Advances in neural information processing systems*, pp. 1090–1098, 2010.

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

Larsen, Anders Boesen Lindbo, Sønderby, Søren Kaae, and Winther, Ole. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015.

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

Liu, Ziwei, Luo, Ping, Wang, Xiaogang, and Tang, Xiaoou. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3730–3738, 2015.

MarcAurelio Ranzato, Christopher Poultney and Chopra, Sumit. Efficient learning of sparse representations with an energy-based model. 2007.

Mathieu, Michael, Couprie, Camille, and LeCun, Yann. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

Pezeshki, Mohammad, Fan, Linxi, Brakel, Philemon, Courville, Aaron, and Bengio, Yoshua. Deconstructing the ladder network architecture. *arXiv preprint arXiv:1511.06430*, 2015.

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

Ranzato, Marc'Aurelio, Boureau, Y-Lan, Chopra, Sumit, and LeCun, Yann. A unified energy-based framework for unsupervised learning. In *Proc. Conference on AI and Statistics (AI-Stats)*, 2007.

Rasmus, Antti, Berglund, Mathias, Honkala, Mikko, Valpola, Harri, and Raiko, Tapani. Semi-supervised learning with ladder networks. In *Advances in Neural Information Processing Systems*, pp. 3546–3554, 2015.

Rifai, Salah, Vincent, Pascal, Muller, Xavier, Glorot, Xavier, and Bengio, Yoshua. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 833–840, 2011.

Russakovsky, Olga, Deng, Jia, Su, Hao, Krause, Jonathan, Satheesh, Sanjeev, Ma, Sean, Huang, Zhiheng, Karpathy, Andrej, Khosla, Aditya, Bernstein, Michael, Berg, Alexander C., and Fei-Fei, Li. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Salimans, Tim, Goodfellow, Ian, Zaremba, Wojciech, Cheung, Vicki, Radford, Alec, and Chen, Xi. Improved techniques for training gans. *arXiv preprint arXiv:1606.03498*, 2016.

Vincent, Pascal, Larochelle, Hugo, Lajoie, Isabelle, Bengio, Yoshua, and Manzagol, Pierre-Antoine. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010.

Vinyals, Oriol, Blundell, Charles, Lillicrap, Timothy, Kavukcuoglu, Koray, and Wierstra, Daan. Matching networks for one shot learning. *arXiv preprint arXiv:1606.04080*, 2016.

Yu, Fisher, Seff, Ari, Zhang, Yinda, Song, Shuran, Funkhouser, Thomas, and Xiao, Jianxiong. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

Zhao, Junbo, Mathieu, Michael, Goroshin, Ross, and Lecun, Yann. Stacked what-where auto-encoders. *arXiv preprint arXiv:1506.02351*, 2015.

A APPENDIX: TECHNICAL POINTS OF SECTION 2.2

Lemma 1. *Let $a, b \geq 0$, $\varphi(y) = ay + b\,[m - y]^+$. The minimum of φ on $[0, +\infty)$ exists and is reached in m if $a < b$, and it is reached in 0 otherwise (the minimum may not be unique).*

Proof. The function φ is defined on $[0, +\infty)$, its derivative is defined on $[0, +\infty)\backslash\{m\}$ and $\varphi'(y) = a - b$ if $y \in [0, m)$ and $\varphi'(y) = a$ if $y \in (m, +\infty)$.
So when $a < b$, the function is decreasing on $[0, m)$ and increasing on $(m, +\infty)$. Since it is continuous, it has a minimum in m. It may not be unique if $a = 0$ or $a - b = 0$.
On the other hand, if $a \geq b$ the function φ is increasing on $[0, +\infty)$, so 0 is a minimum. □

Lemma 2. *If p and q are probability densities, then $\int_x \mathbb{1}_{p(x)<q(x)}\mathrm{d}x = 0$ if and only if $\int_x \mathbb{1}_{p(x)\neq q(x)}\mathrm{d}x = 0$.*

Proof. Let's assume that $\int_x \mathbb{1}_{p(x)<q(x)}\mathrm{d}x = 0$. Then

$$\begin{aligned}
& \int_x \mathbb{1}_{p(x)>q(x)}(p(x) - q(x))\mathrm{d}x && (15)\\
= & \int_x (1 - \mathbb{1}_{p(x)\leq q(x)})(p(x) - q(x))\mathrm{d}x && (16)\\
= & \int_x p(x)\mathrm{d}x - \int_x q(x)\mathrm{d}x + \int_x \mathbb{1}_{p(x)\leq q(x)}(p(x) - q(x))\mathrm{d}x && (17)\\
= & 1 - 1 + \int_x \left(\mathbb{1}_{p(x)<q(x)} + \mathbb{1}_{p(x)=q(x)}\right)(p(x) - q(x))\mathrm{d}x && (18)\\
= & \int_x \mathbb{1}_{p(x)<q(x)}(p(x) - q(x))\mathrm{d}x + \int_x \mathbb{1}_{p(x)=q(x)}(p(x) - q(x))\mathrm{d}x && (19)\\
= & 0 + 0 = 0 && (20)
\end{aligned}$$

So $\int_x \mathbb{1}_{p(x)>q(x)}(p(x) - q(x))\mathrm{d}x = 0$ and since the term in the integral is always non-negative, $\mathbb{1}_{p(x)>q(x)}(p(x) - q(x)) = 0$ for almost all x. And $p(x) - q(x) = 0$ implies $\mathbb{1}_{p(x)>q(x)} = 0$, so $\mathbb{1}_{p(x)>q(x)} = 0$ almost everywhere. Therefore $\int_x \mathbb{1}_{p(x)>q(x)}\mathrm{d}x = 0$ which completes the proof, given the hypothesis. □

Proof of theorem 2 The sufficient conditions are obvious. The necessary condition on G^* comes from theorem 1, and the necessary condition on $D^*(x) \leq m$ is from the proof of theorem 1.
Let us now assume that $D^*(x)$ is not constant almost everywhere and find a contradiction. If it is not, then there exists a constant C and a set $\mathcal{S}$ of non-zero measure such that $\forall x \in \mathcal{S}, D^*(x) \leq C$ and $\forall x \notin \mathcal{S}, D^*(X) > C$. In addition we can choose $\mathcal{S}$ such that there exists a subset $\mathcal{S}' \subset \mathcal{S}$ of non-zero measure such that $p_{data}(x) > 0$ on $\mathcal{S}'$ (because of the assumption in the footnote). We can build a generator G_0 such that $p_{G_0}(x) \leq p_{data}(x)$ over $\mathcal{S}$ and $p_{G_0}(x) < p_{data}(x)$ over $\mathcal{S}'$. We compute

$$\begin{aligned}
U(G^*, D^*) - U(G_0, D^*) &= \int_x (p_{data} - p_{G_0})D^*(x)\mathrm{d}x && (21)\\
&= \int_x (p_{data} - p_{G_0})(D^*(x) - C)\mathrm{d}x && (22)\\
&= \int_{\mathcal{S}} (p_{data} - p_{G_0})(D^*(x) - C)\mathrm{d}x + \int_{\mathcal{R}^N\backslash\mathcal{S}} (p_{data} - p_{G_0})(D^*(x) - C)\mathrm{d}x && (23)\\
&> 0 && (24)
\end{aligned}$$

which violates equation 4.

B APPENDIX: MORE INTERPRETATIONS ABOUT GANS AND ENERGY-BASED LEARNING

TWO INTERPRETATIONS OF GANS

GANs can be interpreted in two complementary ways. In the first interpretation, the key component is the generator, and the discriminator plays the role of a trainable objective function. Let us imagine that the data lies on a manifold. When the discriminator is properly trained, it will discriminate samples on or off the manifold. If the generator is able to produce a sample on the manifold, it gets no gradient thereafter; if the generator produces a sample far away from the manifold, it will get a gradient indicating how to modify its output so it could approach to the manifold. In such scenario, the discriminator acts to punish the generator when it produces samples outside the manifold. This can be understood as a way to train the generator with a set of possible desired outputs (e.g. the manifold) instead of a single desired output as in traditional supervised learning.

For the second interpretation, the key component is the discriminator, and the generator is merely trained to produce contrastive samples. The goal of the discriminator is to find a good energy function to mark out the data manifold in the space. We show that by iteratively and interactively feeding contrastive samples, the generator is able to help the discriminator better capture the data manifold, in section 4.2, which demonstrates the regularization effect the contrastive samples bring about for learning energy function.

C APPENDIX: EXPERIMENT SETTINGS

MORE DETAILS ABOUT THE GRID SEARCH

For training both EBGANs and GANs for the grid search, we use the following setting:

- Batch normalization (Ioffe & Szegedy, 2015) is applied after each weight layers, except for the generator output layer and the discriminator input layer (as in Radford et al. (2015)).
- Training images are scaled into range [-1,1]. Correspondingly the generator output layer is followed by a `Tanh` function.
- The non-parametric `ReLU` is adopted as the non-linearity function throughout architectures.
- We initialize all the weight layers in D from $\mathcal{N}(0, 0.002)$ and in G from $\mathcal{N}(0, 0.02)$. The bias are initialized to be 0.

We evaluate the models from the grid search by calculating a tweaked inception score, $I' = E_x KL(p(y)||p(y|\mathbf{x}))$, where $\mathbf{x}$ denotes a generated sample and y is the label predicted by a MNIST classifier that is trained offline using the entire MNIST training set. Two main changes were made from its original form: (i)-we swap the order of the pair of distributions, i.e., $p(y)$ and $p(y|\mathbf{x})$, in the asymmetrical KL-divergence measure; (ii)- we omit the $e^{(\cdot)}$ operation. Not only that the tweaked score presumably conveys the exact information as its original form, these changes help us shrink the score scale so as to produce more compact histogram. It is also worth noting that although we inherit the name "inception score" from Salimans et al. (2016), the evaluation isn't related to the "inception" model trained on ImageNet. The classifier is a regular ConvNet trained on MNIST.

The generations showed in figure 4 are the best GAN or EBGAN yielding the best inception score from the grid search. Their configurations are:

- figure 4(a): `nLayerG`=5, `nLayerD`=2, `sizeG`=1600, `sizeD`=1024, `dropoutD`=0, `optimD`=SGD, `optimG`=SGD, `lr`=0.01.
- figure 4(b): `nLayerG`=5, `nLayerD`=2, `sizeG`=800, `sizeD`=1024, `dropoutD`=0, `optimD`=ADAM, `optimG`=ADAM, `lr`=0.001, `margin`=10.
- figure 4(c): same as (b), with $\lambda_{PT} = 0.1$.

LSUN & CELEBA

We use a deep convolutional generator analogous to DCGAN's and a deep convolutional auto-encoder for the discriminator. The auto-encoder is composed of strided convolution in the feed-

forward pathway and fractional-strided convolutions in the feedback pathway. We leave the usage of upsampling or switches-unpooling (Zhao et al., 2015) to future research. We also used the bag of guidelines presented in Radford et al. (2015) for training EBGANs. The configuration of the deep auto-encoder is:

- Encoder: `(64)4c2s-(128)4c2s-(256)4c2s`
- Decoder: `(128)4c2s-(64)4c2s-(3)4c2s`

where "`(64)4c2s`" denotes a convolution/deconvolution layer with 64 output feature maps and kernel size 4 with stride 2. The margin m is set to 80 for LSUN and 20 for CelebA.

IMAGENET

We built deeper models in both 128×128 and 256×256 experiments, in a similar fashion to section 4.3,

- 128×128 model:
  - Generator: `(1024)4c-(512)4c2s-(256)4c2s-(128)4c2s-(64)4c2s-(64)4c2s-(3)3c`
  - Noise #planes: `100-64-32-16-8-4`
  - Encoder: `(64)4c2s-(128)4c2s-(256)4c2s-(512)4c2s`
  - Decoder: `(256)4c2s-(128)4c2s-(64)4c2s-(3)4c2s`
  - Margin: 40
- 256×256 model:
  - Generator: `(2048)4c-(1024)4c2s-(512)4c2s-(256)4c2s-(128)4c2s-(64)4c2s-(64)4c2s-(3)3c`
  - Noise #planes: `100-64-32-16-8-4-2`
  - Encoder: `(64)4c2s-(128)4c2s-(256)4c2s-(512)4c2s`
  - Decoder: `(256)4c2s-(128)4c2s-(64)4c2s-(3)4c2s`
  - Margin: 80

Note that we feed noise into every layer of the generator where each noise tensor is initialized into a 4D tensor and subsequently concatenated with current feature maps in the feature space. Such strategy is also employed in Salimans et al. (2016).

D APPENDIX: SEMI-SUPERVISED LEARNING EXPERIMENT SETTING

BASELINE MODEL

As stated in section 4.2, we chose a bottom-layer-cost ladder network as our baseline model. Specifically, we utilize an identical architecture as reported in both papers (Rasmus et al., 2015; Pezeshki et al., 2015); namely a fully-connected network of size `784-1000-500-250-250-250`, with batch normalization and ReLU following each linear layer. To obtain a strong baseline, we tuned the weight of the reconstruction cost with values from the set $\{\frac{5000}{784}, \frac{2000}{784}, \frac{1000}{784}, \frac{500}{784}\}$, while fixing the weight on the classification cost to 1. In the meantime, we also tuned the learning rate with values $\{0.002, 0.001, 0.0005, 0.0002, 0.0001\}$. We adopted Adam as the optimizer with β_1 being set to 0.5. The minibatch size was set to 100. All the experiments are finished by 120,000 steps. We use the same learning rate decay mechanism as the published papers – starting from the two-thirds of total steps (i.e., from step No. 80,000), and linearly decay the learning rate to 0. The result reported in section 4.2 was done by the best tuned setting: $\lambda_{L2} = \frac{1000}{784}, lr = 0.0002$.

EBGAN-LN MODEL

We place the same ladder network architecture into our EBGAN framework and train this EBGAN-LN model the same way as we train the EBGAN auto-encoder model. One trick we found useful was to set the margin to zero while training. The rationale is that the discriminator should to punish the contrastive samples less when they get closer and closer to the data manifold. The extreme case happens when the contrastive samples are pinned exactly on the data manifold, at which time the contrastive samples are "not contrastive anymore", so the discriminator should not pull up the energy on them. Thus, we started training the EBGAN-LN model from the margin value 16 and gradually

annealed it to 0 within the first 60,000 steps. By that time, we observe the reconstruction error of the real image has already reached very low and close to the limitation of the ladder network architecture alone, and what is more, the generated images exhibit good quality. Thereafter we turned off training the generator and kept training the discriminator for 120,000 steps. We set the initial learning rates to be 0.0005 for discriminator and 0.00025 for generator. The other configurations are chosen to be the same as the best baseline LN model. The learning rate starts to decay at step #120,000.

OTHER DETAILS

- We notice that we used the 28×28 version of MNIST dataset in the EBGAN-LN experiment, and used the zero-padded version of size 32×32 for training the EBGAN auto-encoder models. No phenomenal difference has been found due to the zero-padding.
- We generally took the ℓ_2 norm of the discrepancy between input and reconstruction for the loss terms in the EBGAN auto-encoder model as formally described in section 2.1. However, for the EBGAN-LN experiments, we followed the original implementation of ladder networks which used a vanilla form of ℓ_2 loss.
- Borrowed from Salimans et al. (2016), the batch normalization is adopted without the learned parameter γ but merely with a bias term β. It still remains unknown whether such trick could affect learning in some non-ignorable way, so this might have made our baseline model not a strict reproduction of the published models in Rasmus et al. (2015) and Pezeshki et al. (2015).
- All the experiments are conducted upon Torch7. One default feature, `sizeAverage`, is enabled in this series of experiments that the gradient gets divided by the batch size before it is used to embark on backpropogation. This is likely important concerning the λ terms weighing different loss terms and learning rate.

BRIEF ANALYSIS

Ladder network is underutilized when the first or the first few lateral channels unexpectedly carry excessive information so this may loose the regularization effect from the full feedback reconstruction pathway. An extreme case highlighting this is that the first layer copies the entire information from input image (with noise) over to the feedback pathway, and the rest of layers are nullified. This results in making the model no more than a normal denoising auto-encoder. In theory, using the contrastive samples is an effective way to prevent this situation from occurring, because the lateral channels are no longer allowed to blindly copy information.

E APPENDIX: TIPS FOR SETTING A GOOD ENERGY MARGIN VALUE

It is crucial to set a proper energy margin value m in the framework of EBGAN, from both theoretical and experimental perspective. Hereby we provide a few tips:

- Delving into the formulation of the discriminator loss made by equation 1, we suggest a numerical balance between its two terms which concern *real* and *fake* sample respectively. The second term is apparently bounded by $[0, m]$ (assuming the energy function $D(x)$ is non-negative). It is desirable to make the first term bounded in an analogous range. In theory, the upper bound of the first term is essentially determined by (i)-the capacity of D; (ii)-the complexity of the dataset.
- In practice, for the EBGAN auto-encoder model, one can run D (the auto-encoder) alone on the real sample dataset and monitor the loss. When it converges, the consequential loss implies a rough limit on how well such setting of D is capable to fit the dataset, which usually serves as a good start for a hyper-parameter searching on m.
- m being overly large results in a training instability/difficulty, while m being too small is likely to cause the few-mode problem. This property of m is depicted in figure 9.
- One successful technique, as we introduced in appendix D, is to start from a large m and gradually anneal it to 0 along training proceeds. Unlike the feature matching semi-supervised learning technique proposed in Salimans et al. (2016), we show in figure 10 that not only does the EBGAN-LN model achieve a good semi-supervised learning performance, it also produces satisfactory generations.

Abstracting away from the practical experimental tips, the theoretical understanding of EBGAN in section 2.2 also provides some insight for setting a feasible m. For instance, as implied by Theorem 2, setting a large m results in a broader range of γ to which $D^*(x)$ may converge. Instability comes after an overly large γ because it generates two strong gradients pointing to opposite directions, from loss 1, which would demand more finicky optimization settings.

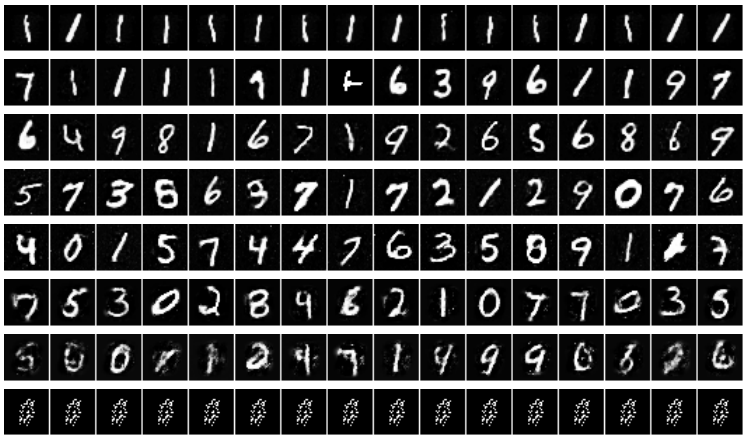

Figure 9: Generation from the EBGAN auto-encoder model trained with different m settings. From top to bottom, m is set to 1, 2, 4, 6, 8, 12, 16, 32 respectively. The rest setting is `nLayerG`=5, `nLayerD`=2, `sizeG`=1600, `sizeD`=1024, `dropoutD`=0, `optimD`=ADAM, `optimG`=ADAM, `lr`=0.001.

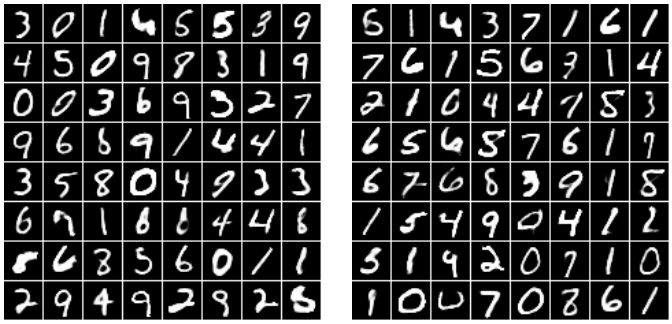

Figure 10: Generation from EBGAN-LN model. The displayed generations are obtained by an identical experimental setting described in appendix D, with different random seeds. As we mentioned before, we used the un-padded version MNIST dataset of image size 28×28 in the EBGAN-LN experiments.

F APPENDIX: MORE GENERATION

LSUN AUGMENTED VERSION TRAINING

For LSUN bedroom dataset, not only we conducted experiment based on the full images, but we also made the experiment grounded on an augmented dataset by cropping patches. All the patches were of size 64×64 cropped from a resized 96×96 full images. The same configuration is utilized as the full-version dataset training. The generation is shown in figure 11.

COMPARISON OF EBGANS AND EBGAN-PTS

To further demonstrate how the pull-away term may benefit EBGAN training, we chose LSUN bedrooms, both full images and its augmented variant (see section 4.3), and CelebA to make further

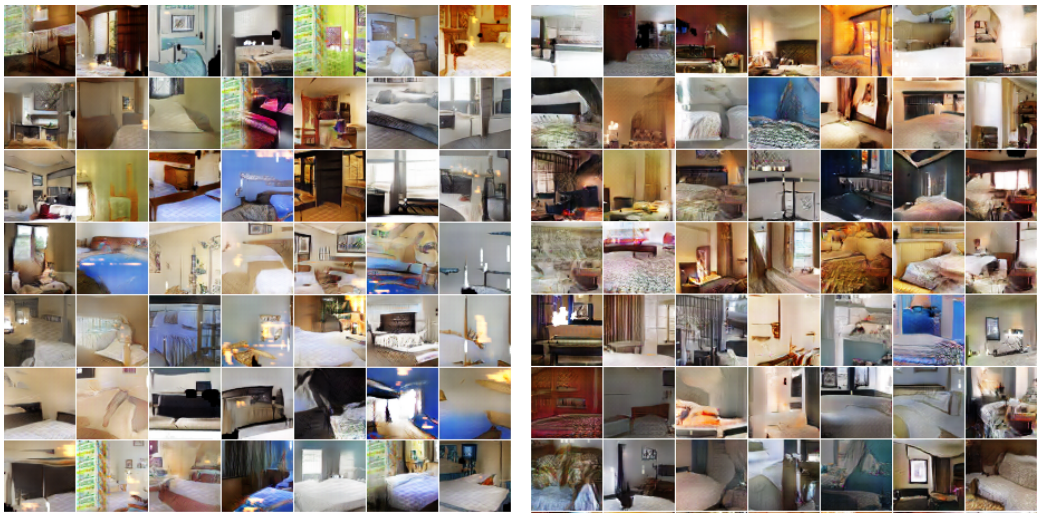

Figure 11: Generation from LSUN bedroom augmented-patches. Left(a): DCGAN generation. Right(b): EBGAN-PT generation.

exploration. Respectively several pairs of EBGAN and EBGAN-PT generations are shown in figure 12, figure 13 and figure 14. Note that all pairs adopt same architectural and hyper-parameter setting as in section 4.3. The weight of the pull-away term is always set to 0.1. It can be concluded that the pull-away term improves both quality and diversity of the model generations.

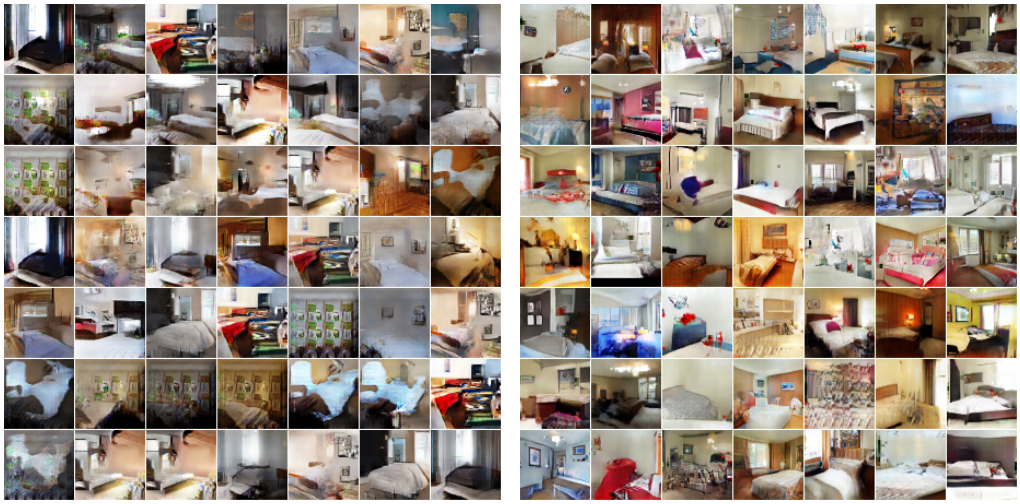

Figure 12: Generation from LSUN bedroom full-images. Left(a): EBGAN. Right(b): EBGAN-PT.

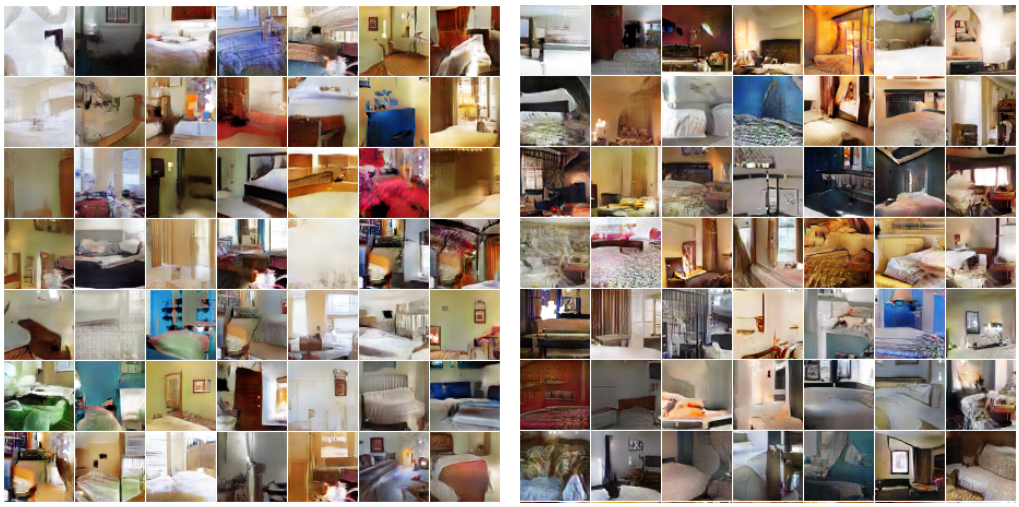

Figure 13: Generation from LSUN augmented-patches. Left(a): EBGAN. Right(b): EBGAN-PT.

Figure 14: Generation from CelebA face dataset. Left(a): EBGAN. Right(b): EBGAN-PT.