
ENERGY-BASED GENERATIVE ADVERSARIAL NETWORK

Junbo Zhao, Michael Mathieu and Yann LeCun

Department of Computer Science, New York University

Facebook Artificial Intelligence Research

{jakezhao, mathieu}@nyu.edu, yann@fb.com

ABSTRACT

We introduce the “Energy-based Generative Adversarial Network” model (EBGAN) which views the discriminator as an energy function that associates low energies with the regions near the data manifold and higher energies with other regions. Similar to the probabilistic GANs, a generator is 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, an instantiation of EBGAN is to use an auto-encoder architecture, with the energy being the reconstruction error. 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

Generative Adversarial Networks (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 samples of a real dataset from “fake” samples produced by the generator. The generator is trained to produce samples that the discriminator cannot distinguish from real data samples. To do so, the generator uses input vectors from an easy-to-sample random source and generates “fake” samples that are fed to the discriminator. During training, the generator “cheats” by receiving the gradient of the output of the discriminator with respect to its input. In the original formulation of GAN in Goodfellow et al. (2014), the discriminator outputs a probability and, under certain conditions, convergence occurs when the distribution produced by the generator matches the data distribution. From an optimization point of view, the convergence of GAN can be seen as reaching a saddle point of an objective function which is minimized with respect to the discriminator parameters and maximized with respect to the generator parameters.

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 (more details are in section 2.3), 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.

As a proof of concept, we perform experiments in which the discriminator is an auto-encoder architecture in which the energy is the reconstruction error. More details on the formulation of EBGAN are provided in the appendix 5.

Our main contributions are summarized as follows:

- (i) An energy-based formulation for adversarial training.
- (ii) An EBGAN framework with the discriminator using an auto-encoder architecture in which the energy is the reconstruction error.
- (iii) 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 parameters and architecture.
- (iv) A repelling regularizer that helps prevent the generator from producing samples in one or a few modes of the data distribution and improves the diversity of generated samples.
- (v) 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 AUTO-ENCODER MODEL

The diagram of an EBGAN with an auto-encoder discriminator is depicted in figure 1. The generator (G) takes random vector z as input, and transform it into a sample $G(z)$, for example an image. The discriminator (D), whose output is a scalar energy, takes either real or generated images, and estimates the energy value E accordingly, where $E \in \mathbb{R}$.

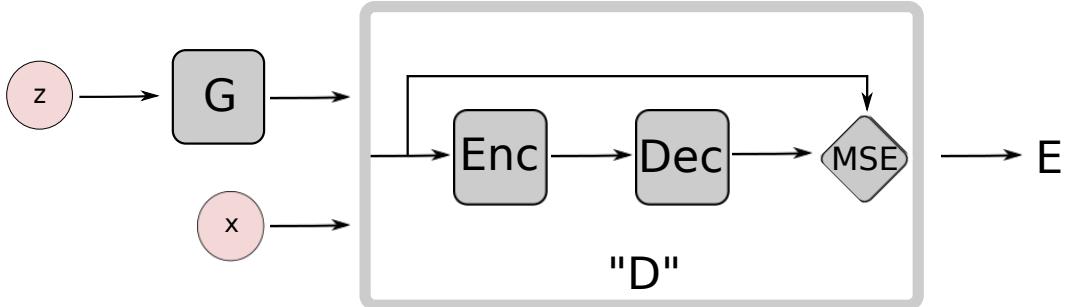


Figure 1: EBGAN architecture.

2.1 OBJECTIVE FUNCTIONALS

The objective functional minimized with respect to the parameters of the discriminator by the learning procedure has the effect of shaping the energy function such that the energy of data samples is low, while the energy of samples coming from the generator is higher. In this work, we chose to use a margin loss, but many other choices are possible LeCun et al. (2006). Given a positive margin m , a data sample x and a generated sample $G(z)$, the loss takes the form:

$$f_D(x, z) = D(x) + [m - D(G(z))]^+ \quad (1)$$

$$= \|Dec(Enc(x)) - x\| + [m - \|Dec(Enc(G(z))) - G(z)\|]^+, \quad (2)$$

where $[\cdot]^+ = \max(0, \cdot)$. This loss is minimized with respect to the parameters of the discriminator (i.e. the parameters of the encoder and the decoder).

Conversely, the generator G is trained to minimize the following objective function:

$$f_G(z) = \|D(G(z))\| \quad (3)$$

$$= \|Dec(Enc(G(z))) - G(z)\| \quad (4)$$

Minimizing $f_G(z)$ with respect to the parameters of G will have the effect of maximizing the second term of the discriminator loss 2. While this remains a form of adversarial training, the resulting optimization cannot be formulated as a gradient-based optimization procedure to find a saddle point of a well-defined objective (unlike the original GAN formulation). It is because the generator loss is no longer the negative form of the discriminator loss in the EBGAN framework; however, with respect to the parameters of G , all the minima of f_G are also maxima of f_D . We hypothesize that this is sufficient to ensure a suitable fixed point of the adversarial optimization procedure.

2.2 REPELLING REGULARIZER

A common problem with GAN framework is that the generator tends to only generate samples that are clustered in one or a few modes of the regions of high data density, instead of spanning the whole range. One method called “minibatch discrimination” (MBD) was recently proposed to mitigate this problem (Salimans et al., 2016). The idea is to provide an entire of batch of samples at a time to the discriminator, instead of just one. The discriminator will then easily identify the pattern as fake if they are all similar to each other. This drives the generator to produce diverse samples within the minibatch.

MBD is difficult to implement with energy-based discriminators that utilize auto-encoder architectures. We propose an alternative technique to maintain the diversity of the samples. Many unsupervised learning methods involving auto-encoders put a regularizer on the hidden representation, for example a sparsity term (MarcAurelio Ranzato & Chopra, 2007; Kavukcuoglu et al., 2010; Ng). In the present work we propose a “repelling regularizer” or a pull-away term (PT) imposed on the encoder representation $S = \text{Enc}(G(z))$. We will use “EBGAN-PT” to refer to this variant. The pull-away term is defined as follows,

$$f_{PT}(S) = \frac{1}{bs(bs-1)} \sum_i \sum_{j \neq i} \left(\frac{S_i^T S_j}{\|S_i\| \|S_j\|} \right)^2,$$

where bs is the batch size and S_i is the i -th sample in the minibatch representation S . This term decreases the magnitude of cosine similarity between the sample representations within the minibatch and thus makes them as mutually orthogonal as possible. When the gradient of this term is back-propagated to the generator, it causes it to avoid producing identical or repeating samples and thus increases their diversity. Note that the pull-away term is used in the generator loss but not in the discriminator loss. The rationale for choosing the cosine instead of the Euclidean distance is to make the term bounded below and invariant to scale. In the experiments, the pull-away term is given a weight of **0.1** and added to the generator loss.

2.3 TRAINING STABILITY

We assess the training stability problem for GANs by examining the loss functional of the discriminator. The negative log-likelihood loss function used in regular GAN training originates from a probabilistic formulation, and it can be bridged with the energy-based view by applying the Gibbs distribution (LeCun et al., 2006):

$$P(Y|X) = \frac{e^{-\beta E(Y,X)}}{\sum_{Y \in \mathcal{Y}} e^{-\beta E(Y,X)}}, \quad (5)$$

where $E(Y, X)$ is assigned to be the corresponding energy value for tuple (Y, X) , β is a hyper-parameter and \mathcal{Y} defines a set of possible labels for Y (in the regular GAN setting, it contains “real” and “fake”). The negative log-likelihood loss pushes toward a solution where $P(Y^*|X) = 1$ when Y^* is the desired label matches X . This, based on equation 5, is attained if and only if $E(Y^*, X) \rightarrow -\infty$ because the denominator contains the numerator. That being said, the discriminator in a regular GAN can be explained as an energy function trained with a margin loss which pushes the energy gap between regions near the data manifold (positive samples) and regions far away from the manifold (negative samples) to be ∞ . We conjecture that such extreme setting of the energy margin may excessively enlarge the difference between regions on and off the manifold, leading to a difficulty for training GANs. By contrast, as explained in section 2.1, the general EBGAN framework (not limited to using auto-encoders and reconstruction error) can be trained with a finite energy margin.

2.4 COMMON CONCERNS FOR TRAINING AUTO-ENCODERS AND THE USE OF L2 LOSS

One common concern in training auto-encoders is the model learns little more than an identity function. By adding regularizers, several techniques were addressed to resolve such problem (Vincent et al., 2010; Rifai et al., 2011; Ng; MarcAurelio Ranzato & Chopra, 2007; Kavukcuoglu et al., 2010). However, we claim that the EBGAN framework is free from such concern. On the surface, a universal identity function won't be able to satisfy the objective of the energy function D , because it is not only required to learn to reconstruct real samples, but is also expected to disorder the reconstruction from the generated samples. At the heart of EBGANs, the auto-encoders exclusively serve as a specific form of energy function, as being used to discern the real and fake samples by delivering an energy value. Learning an identity function on the real samples while disordering fake samples' reconstruction is very acceptable in the EBGAN framework, so long as it exerts certain discrimination capacity.

The same rationale is also applied to the usage of L2 loss, which was considered as being sub-optimal (Mathieu et al., 2015) in certain tasks due to its regression-to-mean problem. However, we argue that obtaining sharp reconstructions is not what EBGANs aim at. We also empirically found that increasing the margin value will coarsen the reconstruction of generated and real images altogether, while the generation is able to maintain its quality.

In a word, within EBGAN framework, many energy formulations are widely applicable in place of D . The choice of auto-encoders, whether regularized or unregularized, with L2 loss are deemed as one formulation among all the potentials in LeCun et al. (2006).

3 RELATED WORK

Our work primarily bridges GANs with energy-based models. Despite that we chose a plain auto-encoders with L2 loss as our discriminator or energy function, a variety of models, in theory, can be plugged in seamlessly, such as Carreira-Perpinan & Hinton (2005), Gutmann & Hyvärinen (2010), Vincent et al. (2010) and Rifai et al. (2011).

In quest for better techniques for stabilizing GAN training, several papers was presented with helpful techniques in 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 pioneering for root investigating GANs' instability problem on a model level.

EBGANs also bear some similarity with Larsen et al. (2015), in which the authors managed to train a GAN as a replacement of L2 criterion module. EBGANs nonetheless are built from a different perspective aiming at using auto-encoders to stabilize GAN training itself. Another energy-based model was recently proposed by Kim & Bengio (2016). Such model is potentially connected with both GANs and EBGANs, whereas the method is not devised to use adversarial training paradigm.

4 EXPERIMENTS

4.1 COMPARISON WITH GANS BY EXHAUSTIVE GRID SEARCH

To prove a better training stability possessed by EBGANs than GANs may demand a large number of experiments validating a complete set of architectural parameters and training settings. Therefore, we devised a pair of exhaustive grid search comparisons. Since the search space involving the possible meta-parameters folded with hyper-parameters is colossal, in order to make the experiment more viable, we restricted our grid search workhouse for both EBGANs and GANs being fully-connected and trained on MNIST. The convolutional architectures applied on larger and more complex datasets are exhibited in the following sections.

The architectural parameters include:

- number of layers in G , nLayerG
- number of layers in D , nLayerD
- number of neurons in G , sizeG
- number of neurons in D , sizeD

Table 1: Grid search specs

Settings	EBGANs	GANs
nLayerG	[2, 3, 4, 5]	[2, 3, 4, 5]
nLayerD	[2, 3, 4, 5]	[2, 3, 4, 5]
sizeG	[400, 800, 1600, 3200]	[400, 800, 1600, 3200]
sized	[128, 256, 512, 1024]	[128, 256, 512, 1024]
dropoutD	[true, false]	[true, false]
optimD	adam	[adam, sgd]
optimG	adam	[adam, sgd]
lr	0.001	[0.01, 0.001, 0.0001]
Total number of experiments:	512	6144

- if using dropout in D , `dropoutD`

We further expand the grid by adding optimization method and learning rate in the search of a better GAN:

- using Adam (Kingma & Ba, 2014) or SGD in D , `optimD`
- using Adam or SGD in G , `optimG`
- learning rate, `lr`

Note that we additionally impose the following restriction on EBGANs:

- using learning rate 0.001 and Adam for both part
- `nLayerD` counts for a summation of layers together for Enc and Dec . Also for simplicity, we fix the number of Dec to **1** and just tune the encoder layers
- the margin is set to **10** throughout all experiments of EBGANs

Batch normalization (Ioffe & Szegedy, 2015) is applied generally as a default setting. ReLU, except for the last layer uses hyperbolic tangent Tanh, is adopted as the non-linearity function throughout architectures. For both EBGANs and GANs, 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 all 0. The specific grid search setting is shown in table 1. Overall, we have trained **6144** GAN models and **512** EBGAN models.

Evaluation It is impossible to rely on human effort looking over all displayed generations for every experiment out of such a giant grid. We chose to use a recently proposed numerical assessment approach “inception-score” (Salimans et al., 2016),

$$I = E_x KL(p(y|\mathbf{x})||p(y)), \quad (6)$$

where \mathbf{x} denotes one generated sample and y is the label predicted by a MNIST classifier trained offline using the entire MNIST training set. Notice we ignore the exponentiating from the original formula for brevity.

Histograms We start by accessing the results by plotting histograms in figure 2(a). Alongside these general scores, we also plot the histogram of the models under certain `nLayerG` and `nlayerD` cap in figure 2(b,c). As we use the portion in percentage for each bin as the visual feature in the histograms, one may argue that 12 times more experiments may put GANs in an unfair position because the sensitive setting of the optimization method and learning rate may make certain sub-grid completely fail. Therefore, we have separated out the optimization related setting from GANs grid; they are `optimD`, `optimG` and `lr`. As such, the grid of GANs is divided into $2 \times 2 \times 3 = 12$ different combinations of these three features and each of them is therefore associated with a sub-grid consisting of **512** experiments individually, which equals to the number of EBGAN experiments. We plot the histograms for all 12 sub-grids accordingly, along with the EBGANs’ score as a benchmarking comparison, in figure 3. Note that the histogram comparison only involves EBGANs trained **without** the pull-away term.

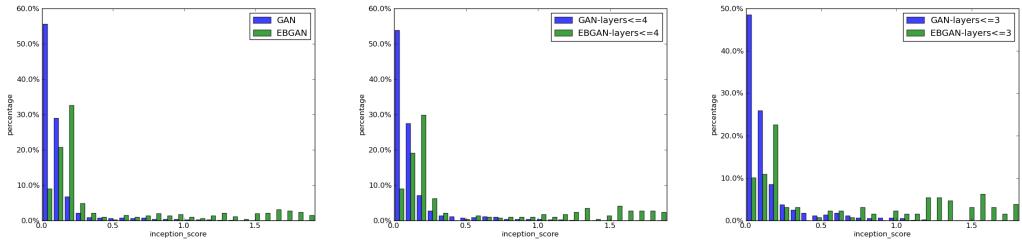


Figure 2: **(Zooming in on pdf file is recommended.)** Histogram of the inception scores from the grid search. The x-axis carries the inception score I and y-axis informs the portion of the models (in percentage) falling into certain bins. Left (a): general comparison of EBGANs against GANs; Middle (b):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

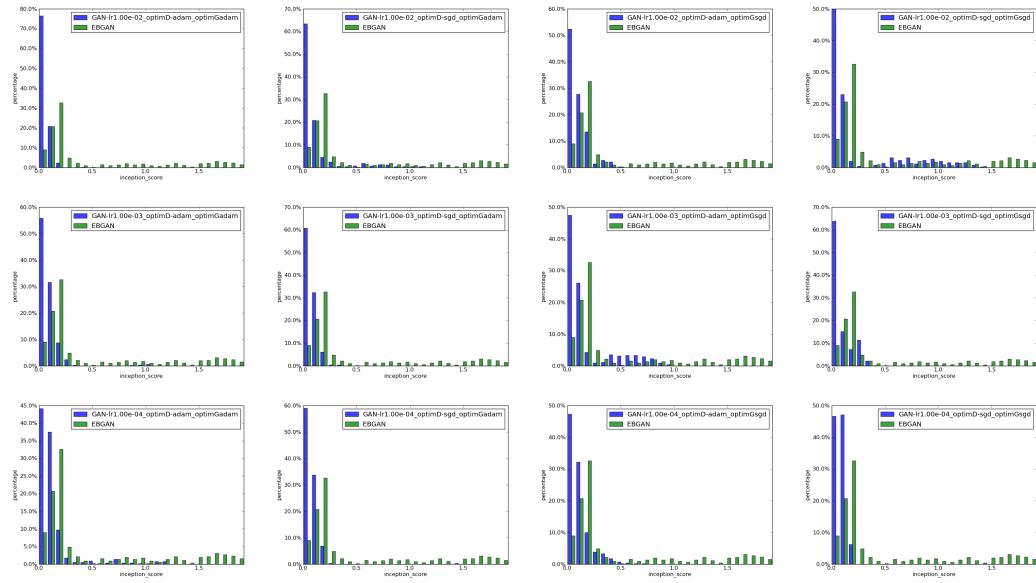


Figure 3: **(Zooming in on pdf file is recommended.)** Histogram of the inception scores grouped by different optimization combinations, drawn from `optimD`, `optimG` and `lr` (See text).

Both figures together show that EBGANs consistently perform more reliably and also satisfactorily than GANs, regarding less finicky architectural and parameter requirements. It can be concluded that training GANs may demand exquisitely tuned hyper-parameters and meta-parameters, meaning that much narrower range and less parameters could possibly get GANs to work, compared to EBGANs.

A few more lessons can be learned from those 12 separated histogram plots in figure 3. Using SGD on both D and G results in the most stable result in training GANs (figure 3 first-row fourth-column); yet, it still lags much behind EBGANs. For the other settings, some combinations occasionally reap acceptable results, such as using Adam for both and adopting a smaller learning rate $1e-04$ shown in figure 3 third-row first-column. However, a number of settings hardly work even once.

Heatmaps To better penetrate the results pool, we further investigate the best and median performance among the models, and plot them into heatmaps in figure 4(a) and figure 4(b) respectively. Statistically, the median value reflects the typical performance level. Meanwhile the best performance of the model is revealed by the maximum inception score amongst different settings.

In this regard, the best-performance heatmaps in figure 4(a) apparently show a superior advantage of EBGAN framework, with or without the pull-away term, upon ordinary GANs. For the median-performance heatmaps in figure 4(b), we see the most robust performance belongs to EBGANs. We

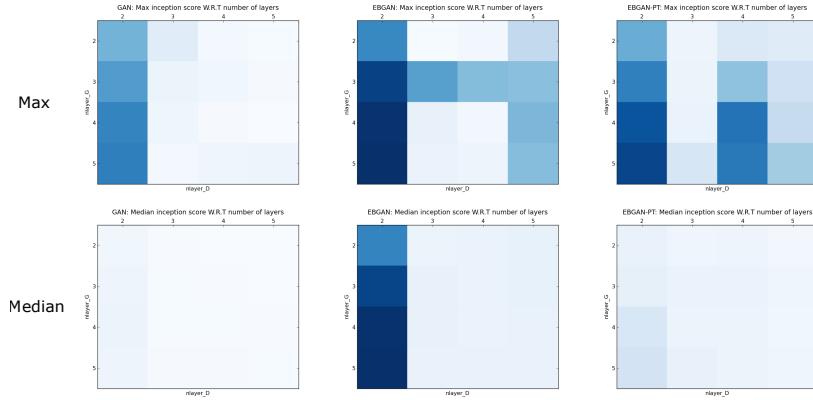


Figure 4: Upper(a): Maximum inception score heatmaps. Lower(b): Median inception score heatmaps. From left to right, the plots are ordered by GANs, EBGANs, followed by EBGAN-PTs. The horizontal axis denotes $nlayer_D$ and the vertical axis denotes $nlayer_G$. The colors are drawn based on the (max or median) inception score accordingly among the sub-grid experiments delimited by layer $nlayer_{[GD]}$. Note each group of the three maps have adopted a uniform color intensity palette, and two different palettes are used respectively for max and median heatmaps.

argue that the performance of EBGAN-PT models were impaired due to a fixed weight for the pull-away term (**0.1**) was adopted regardless of the model setting. We further demonstrate the appealing effect that the pull-away term brings about via generation in section 4.2 and appendix 5.

4.2 MNIST

Based on the grid search and its quantitative evaluation results, we chose the best models respectively from ordinary GANs, EBGANs and EBGAN-PTs, and display their generations, in figure 5. While they all generate plausible digits, EBGANs and EBGAN-PTs not only have produced more visually appealing digits, but also perform very reliably, that is, they rarely generate off-manifold digits. Moreover, EBGAN-PT generation in figure 5(c) provides more diversity concerning the aspects including the stylish of the digits and thickness of stroke, compared to the EBGAN generation in figure 5(b).

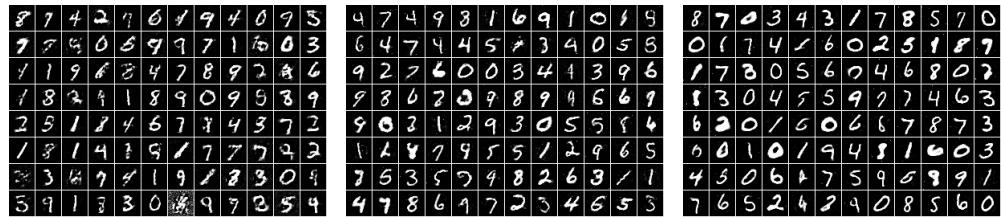


Figure 5: Exemplar generation from the grid search on MNIST. Left(a): Best GAN generation; Middle(b): Best EBGAN generation. Right(c): Best EBGAN-PT generation.

4.3 LSUN & CELEBA

To access the capacity of generating 64×64 RGB images on the basis of using deep convolutional models, we have trained EBGANs on the LSUN bedroom dataset (Yu et al., 2015) and the large scale face dataset CelebA under alignment (Liu et al., 2015). For LSUN bedroom dataset, not only we conducted experiment based on the full images, but we also made the experiment grounded in an augmented dataset by cropping patches. All the patches were of size 64×64 cropped from a resized 96×96 full images.

As it is stated, we use a deep convolutional generator analogous to DCGANs (Radford et al., 2015) and a deep convolutional auto-encoders as the discriminator. Inside the auto-encoders, strided con-

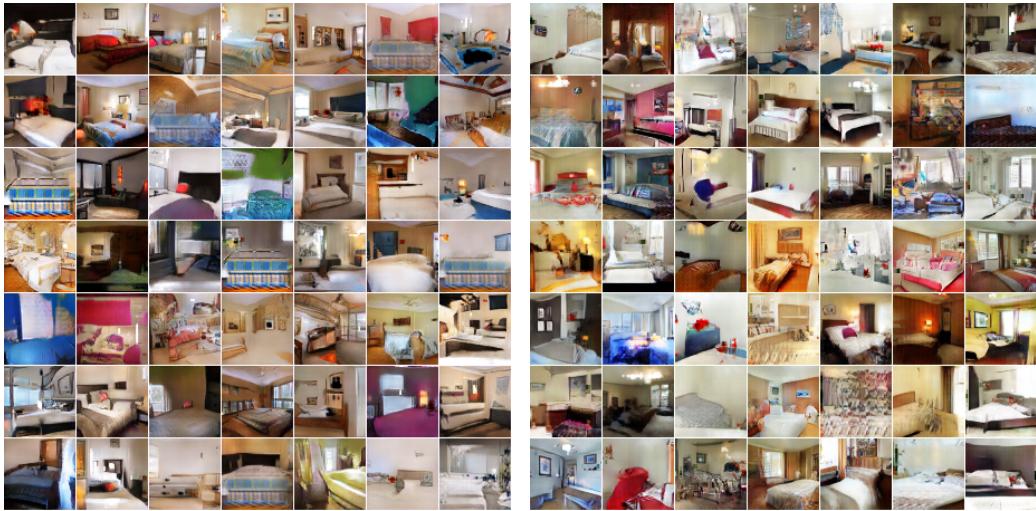


Figure 6: Generation from LSUN bedroom full-images. Left(a): DCGAN generation. Right(b): EBGAN-PT generation.

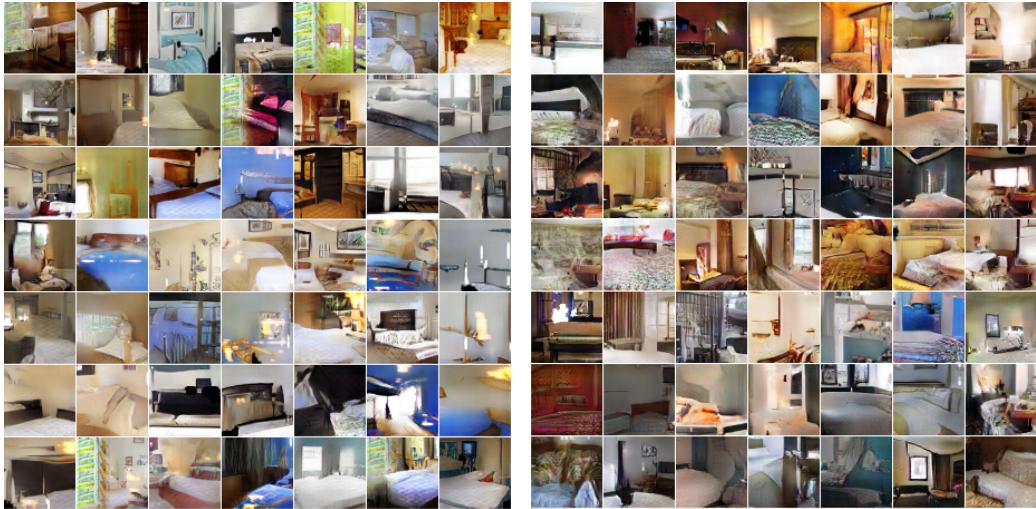


Figure 7: Generation from LSUN bedroom augmented-patches. Left(a): DCGAN generation. Right(b): EBGAN-PT generation.

volution is used in the feedforward pathway and the fractional-strided convolutions are used in the feedback pathway. We leave the usage of upsampling or switches-unpooling (Zhao et al., 2015; Zhang et al.) to future research. 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 of LSUN is set to 80 and of CelebA to 20. We also used the bag of guidelines presented with DCGANs in Radford et al. (2015), for training EBGANs.

In light of comparing EBGANs with DCGANs, we train an DCGAN under the same configuration as the EBGAN-PT model. The LSUN bedroom generations by both frameworks are displayed in figure 6(a,b), with the augmented dataset generation in figure 7(a,b). While both frameworks have produced high quality generations using LSUN bedroom pictures, EBGAN-PTs are shown to response to richer modes and successfully exclude some recurring off-manifold parts, such as the

blue “bed-alike” item appearing frequently at the bottom of the generations in figure 6(a) or the “curtain-alike” item in figure 7(a).

Likewise, CelebA generations of an EBGAN-PT and an identically configured DCGAN are shown in figure 8. The EBGAN-PT model manages to obtain higher quality and more diverse generations, and avoid emitting recurring off-manifold parts.

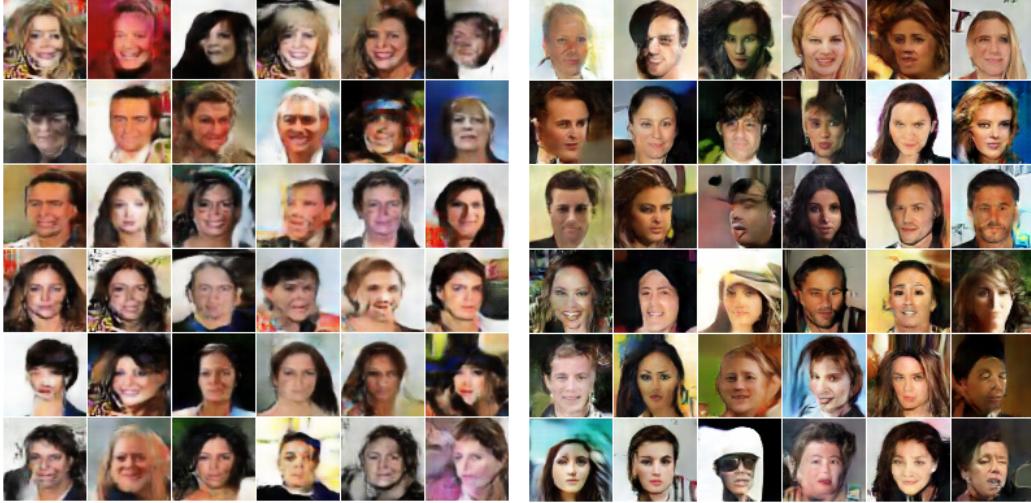


Figure 8: Generation from CelebA face dataset. Left(a): DCGAN generation. Right(b): EBGAN-PT generation.

4.4 IMAGENET

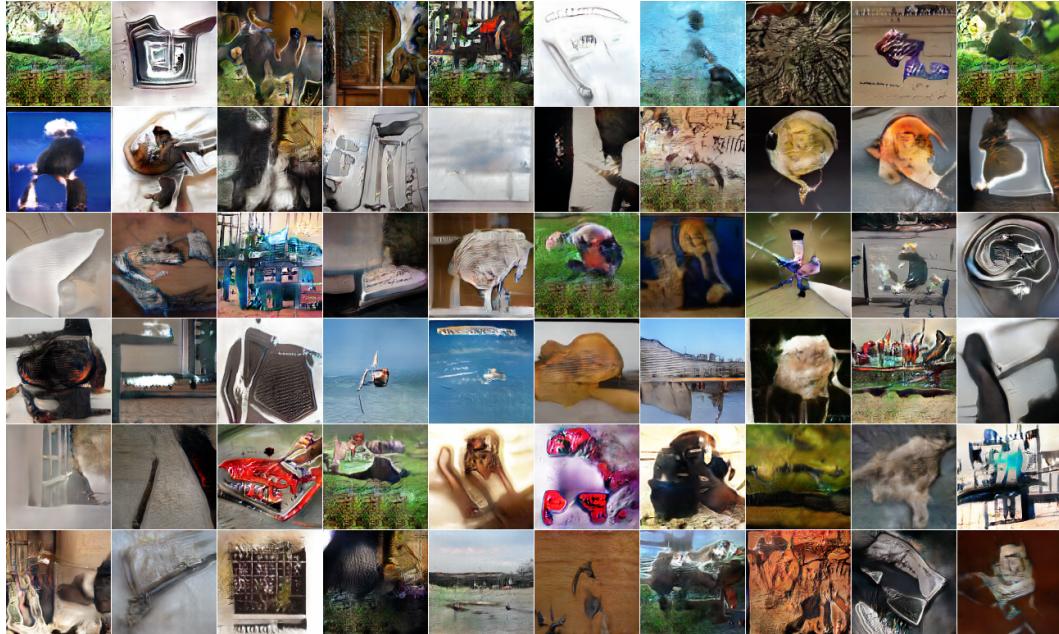


Figure 9: ImageNet 128×128 generations using an EBGAN-PT.

The final step of evaluating EBGANs arrives at generating ImageNet images (Russakovsky et al., 2015). We consider the generation on both 128×128 and 256×256 resolution level. Compared upon the datasets we have shown so far, no doubt has ImageNet assembled an extensively larger and wilder



Figure 10: ImageNet 256×256 generations using an EBGAN-PT.

space so that modeling the data distribution from a generative aspect becomes very challenging. We devised an experiment on 128×128 using full ImageNet-1k dataset, which contains roughly 1.3 million images from 1000 different categories. One step further, we attempted to generate images on a unprecedented resolution level 256×256 . To make the task more viable, we exploit only a dog-breeds subset of ImageNet, following the WordNet IDs provided in Vinyals et al. (2016).

We built deeper models in both 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 blob and subsequently concatenated with current feature maps in the feature space. Such strategy is also employed in Salimans et al. (2016). The results are shown in figure 9 and figure 10 respectively for 128×128 and 256×256 resolution levels. Despite the difficulty of generating images on a high-resolution level, it is manifest that an EBGAN is able to learn about the fact that objects appearing in the foreground, together with various background components resembling grass texture, sea under the horizon, mirrored mountain in the water and buildings etc. In addition, embodied by 256×256 dog-breeds generations, EBGANs absorb the knowledge of dogs such as their body, furs and eyes.

5 OUTLOOK

We bridge two classes of unsupervised learning methods – GANs and auto-encoders – and renovate GAN framework from an alternative energy-based view to the original probabilistic perspective (Goodfellow et al., 2014). EBGANs show better convergence pattern and scalability to generate on large scales. A family of energy-based loss functionals presented in LeCun et al. (2006) are allowed to be incorporated easily into the EBGAN framework. For the future work, the conditional setting is very worth to explore (Denton et al., 2015; Mathieu et al., 2015). Besides, a more formal and exhaustive theoretical understanding, for EBGANs and GANs altogether, is on demand.

In a word, we hope the future research could 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 and Ross Goroshin for insightful discussions. We also thank Emily Denton and Aditya Ramesh for their help with the manuscript.

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.
- Gutmann, Michael and Hyvärinen, Aapo. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*, volume 1, pp. 6, 2010.
- Im, Daniel Jiwong, 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.
- Kim, Taesup and Bengio, Yoshua. Deep directed generative models with energy-based probability estimation. *arXiv preprint arXiv:1606.03439*, 2016.
- 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 Poulton and Chopra, Sumit. Efficient learning of sparse representations with an energy-based model. 2007.
- Mathieu, Michael, Couprise, Camille, and LeCun, Yann. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.
- Ng, Andrew. Sparse autoencoder.
- 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.
- 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.
- Zhang, Yuting, Lee, EDU Kibok, Lee, EDU Honglak, and EDU, UMICH. Augmenting supervised neural networks with unsupervised objectives for large-scale image classification.
- Zhao, Junbo, Mathieu, Michael, Goroshin, Ross, and Lecun, Yann. Stacked what-where auto-encoders. *arXiv preprint arXiv:1506.02351*, 2015.

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.

CONTRASTIVE SAMPLES

One of the critical issues with energy-based learning is how to make the energy function take high values outside of regions of high data density. In GAN and its variants, the generator can essentially be seen as a “smart” way to produce contrastive samples, to which the discriminator ought to give high energy values. On the same direction of producing contrastive samples, a number of approaches were proposed such as the use of noisy samples (Vincent et al., 2010) and noisy gradient descent methods such as contrastive divergence (Carreira-Perpinan & Hinton, 2005). In this regard, we can postulate that the differences between such approaches and the generator in GAN framework include: (i)-GAN’s generator is fully trainable; (ii)-the adversarial training paradigm enables a direct interaction between the processes of producing contrastive samples and learning the energy function.

APPENDIX: 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 exploration. Respectively several pairs of EBGAN and EBGAN-PT generations are shown in figure 11, figure 12 and figure 13. 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.

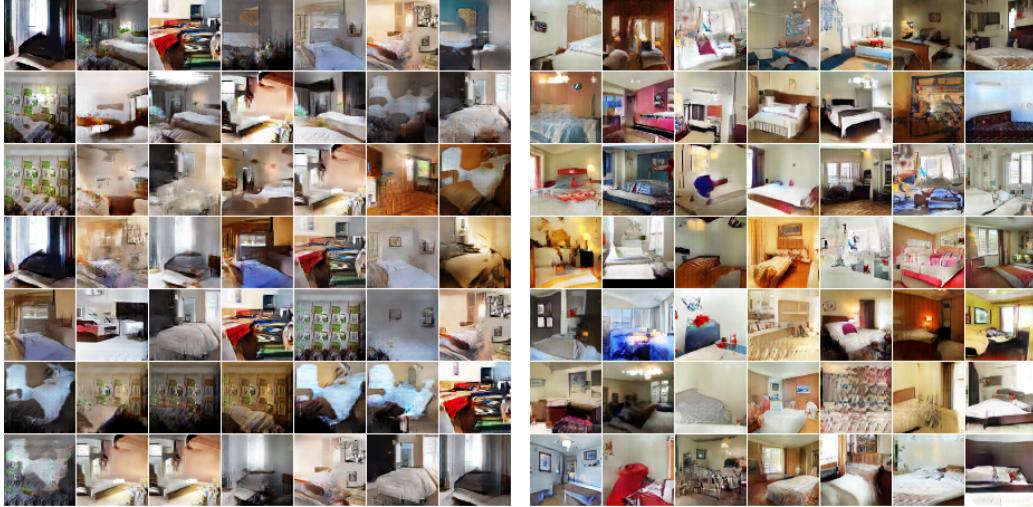


Figure 11: Generation from LSUN bedroom full-images. Left(a): EBGAN generation. Right(b): EBGAN-PT generation.

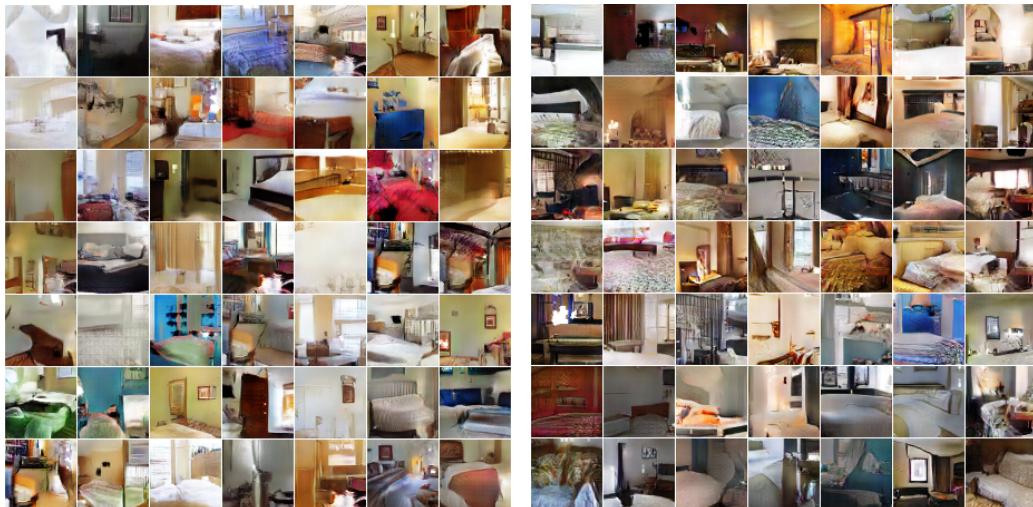


Figure 12: Generation from LSUN bedroom augmented-patches. Left(a): EBGAN generation. Right(b): EBGAN-PT generation.

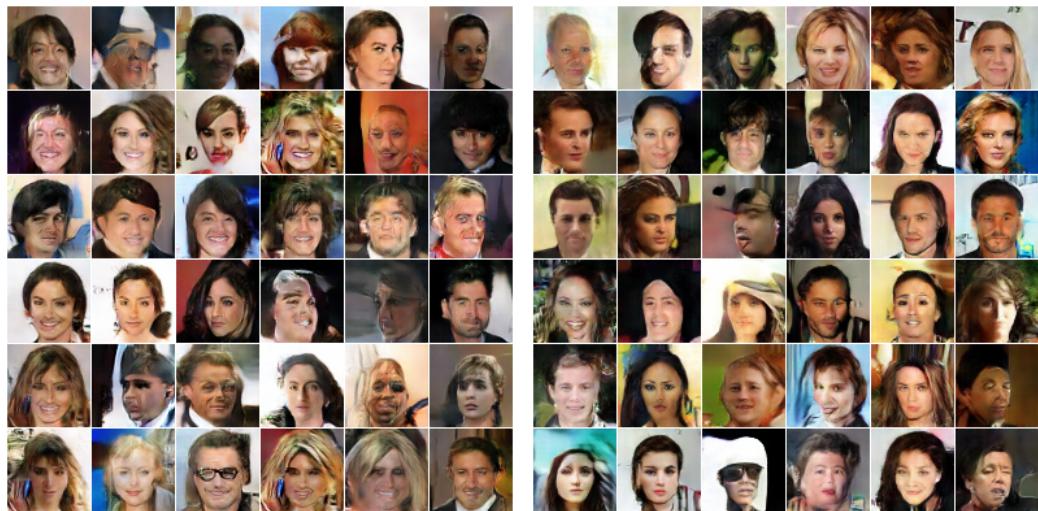


Figure 13: Generation from CelebA face dataset. Left(a): EBGAN generation. Right(b): EBGAN-PT generation.